



ENIAC - 2017

XIV Encontro Nacional de Inteligência Artificial e Computacional

Realização:



Organização:



Promoção:





XIV Encontro Nacional de Inteligência Artificial e Computacional
2 a 5 de outubro de 2017
Uberlândia, MG, Brasil

Anais

Comissão Especial de Inteligência Artificial da Sociedade Brasileira de Computação
(CEIA-SBC)

Comissão Especial de Inteligência Computacional da Sociedade Brasileira de
Computação (CEIC-SBC)

COORDENADORES DO COMITÊ DE PROGRAMA

Aline Paes (UFF)
André Britto (UFS)

EDITORES

Aline Paes (UFF)
André Britto (UFS)
Gina M. B. de Oliveira (UFU)

REALIZAÇÃO

Sociedade Brasileira de Computação

ORGANIZAÇÃO

Faculdade de Computação - Universidade Federal de Uberlândia

Apresentação

O ENIAC 2017 é o décimo quarto evento de uma série de encontros bem-sucedidos em Inteligência Artificial e Inteligência Computacional. O ENIAC teve edições anteriores realizadas em Brasília-DF (1997), Rio de Janeiro-RJ (1999), Fortaleza-CE (2001), Campinas-SP (2003), São Leopoldo-RS (2005), novamente Rio de Janeiro-RJ (2007), Bento Gonçalves-RS (2009), Natal-RN(2011), Curitiba-PR (2012), Fortaleza-CE (2013), São Carlos-SP (2014), Natal-RN (2015) e Recife-PE(2016).

Em 2017, o ENIAC foi realizado em Uberlândia-MG, como parte da sexta Brazilian Conference on Intelligent System (BRACIS'2017). Este ano, também ocorreram em conjunto ao BRACIS'2017 o 5º Symposium on Knowledge Discovery, Mining and Learning (KDMILE'2017), o 32º Simpósio Brasileiro de Banco de Dados (SBBDD'2017) e o Brazilian Symposium in Information and Human Language Technology (STIL'2017). O ENIAC é um evento que oferece um fórum para pesquisadores, profissionais, educadores e estudantes apresentarem e discutirem as inovações, tendências, experiências e evolução nos campos de Inteligência Artificial e Computacional.

Duzentos e sete trabalhos completos foram submetidos para esta edição do ENIAC. Os trabalhos foram revisados por três membros do comitê de programa ou por avaliadores designados pelo comitê, pertencentes a diversas regiões do país. Nestes anais constam os oitenta e quatro trabalhos selecionados apresentados nas sessões técnicas do evento. Além das sessões técnicas a programação do evento também incluiu, em conjunto com o BRACIS 2017, uma palestra e um tutorial ministrados pelo Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho do ICMC-USP.

Agradecemos inicialmente aos autores dos artigos submetidos e em particular, parabenizamos aqueles que foram selecionados para apresentação no evento. Agradecemos especialmente aos membros do comitê de programa e aos revisores por eles designados. Agradecemos também ao Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho pela importante participação na palestra e tutorial. Por fim, agradecemos a todos os que diretamente ou indiretamente nos auxiliaram na organização do ENIAC 2017, em especial aos organizadores do BRACIS 2017, pelo apoio e infraestrutura disponibilizada.

Uberlândia, 04 de outubro de 2017

Aline Paes

André Britto

Coordenadores de Comitê do Programa

Comitê Técnico

Coordenadores de comitê de programa

Aline Paes Universidade Federal Fluminense

André Britto Universidade Federal de Sergipe

Coordenadora geral do BRACIS 2017

Gina M. B. de Oliveira Universidade Federal de Uberlândia

Comitê de programa

Adolfo Neto	Federal University of Technology - Paraná (UTFPR)
Adriano Oliveira	Universidade Federal de Pernambuco
Adriano Werhli	Universidade Federal do Rio Grande
Adrião Duarte Dória Neto	Universidade Federal do Rio Grande do Norte
Ahmed Esmin	Departamento de Ciência da Computação - UFLA
Alessandro Bicho	Federal University of Rio Grande (FURG)
Alessandro Koerich	Pontifícia Universidade Católica do Paraná
Alexandre Delbem	ICMC-USP
Alexandre Plastino	Universidade Federal Fluminense
Aline Villavicencio	Universidade Federal do Rio Grande do Sul
Alvaro Moreira	Universidade Federal do Rio Grande do Sul
Amilcar Cardoso	Universidade de Coimbra
Ana Bazzan	UFRGS
Ana Carolina Lorena	Universidade Federal de São Paulo
Ana Teresa Martins	Universidade Federal do Ceará
André Ponce de Leon F de Carvalho	ICMC-USP/S.Carlos

André Renato Villela da Silva	Universidade Federal Fluminense
Andreia Malucelli	PUCPR
Anisio Lacerda	UFMG
Anne Canuto	UFRN
Antonio Branco	Universidade de Lisboa
Antonio Carlos da Rocha Costa	UFRGS-FURG
Aurora Pozo	UFPR
Bianca Zadrozny	IBM Research Brazil
Bruno Lopes	Universidade Federal Fluminense
Carla Delgado	Universidade Federal do Rio de Janeiro (UFRJ)
Carlos Lopes	Universidade Federal de Uberlândia
Carlos Thomaz	Centro Universitário da FEI
Carmelo Bastos-Filho	University of Pernambuco
Carolina Paula de Almeida	State University in the Midwest of Paraná - UNICENTRO
Cleber Zanchettin	Universidade Federal de Pernambuco
Cristina N. Vasconcelos	Universidade Federal Fluminense (UFF)
David Martins-Jr	Universidade Federal do ABC
Diana Adamatti	Universidade Federal do Rio Grande - FURG
Díbio Leandro Borges	University of Brasília
Edson Matsubara	Fundação Universidade Federal de Mato Grosso do Sul
Emerson Paraíso	Pontificia Universidade Católica do Paraná
Eugênio Oliveira	Faculdade Engenharia Universidade do Porto
Fabio Okuyama	IFRS - Campus Porto Alegre
Fabrcio Enembreck	Pontifical Catholic University of Paraná - PUCPR

Fábio Cozman	USP - Politécnica
Flavia Barros	Universidade Federal de Pernambuco
Flavia Bernardini	Universidade Federal Fluminense
Flavio Tonidandel	Centro Universitário da FEI
Flávio Soares Corrêa da Silva	Universidade de São Paulo
Flávio Varejão	Universidade Federal do Espírito Santo
Francisco De Carvalho	Centro de Informática - CIn/UFPE
George Cavalcanti	Universidade Federal de Pernambuco
Gerson Zaverucha	Federal University of Rio de Janeiro (UFRJ)
Gina Oliveira	Universidade Federal de Uberlândia
Graçaliz Dimuro	Universidade Federal do Rio Grande
Guilherme Barreto	Universidade Federal do Ceará
Gustavo Batista	Universidade de São Paulo
Gustavo Giménez-Lugo	UTFPR
Helder Coelho	Universidade de Lisboa
Heloisa Camargo	UFSCar
Huei Lee	Universidade Estadual do Oeste do Paraná
Inês Dutra	Universidade do Porto
Ivandr� Paraboni	USP Leste
Jo�o Marcos	UFRN
Jo�o Carlos Pereira da Silva	Universidade Federal do Rio de Janeiro
Jo�o Alc�ntara	Universidade Federal do Cear�
Jo�o Balsa	Fac. Ci�ncias da Universidade de Lisboa
Jomi H�bner	UFSC
Jorge Kanda	Universidade Federal do Amazonas

José Augusto Baranauskas	Universidade de São Paulo
José Manoel de Seixas	UFRJ
José Viterbo Filho	Universidade Federal Fluminense
Julio Nievola	PUCPR
Karina Valdivia-Delgado	University of São Paulo
Kate Revoredo	Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Leliane Nunes de Barros	Universidade de São Paulo
Li Weigang	University of Brasília
Liang Zhao	Universidade de São Paulo
Lilian Berton	University of São Paulo
Luís Correia	Universidade de Lisboa
Luis Martí	Universidade Federal Fluminense (UFF)
Luiz Satoru Ochi	Instituto de Computação - Universidade Federal Fluminense
Luiza Mourelle	State University of Rio de Janeiro
Marcelo Ladeira	Universidade de Brasília
Marcilio de Souto	Université d'Orléans
Marco Goldberg	UFRN
Marcos Quiles	Universidade Federal de São Paulo
Marcos Eduardo Valle	Universidade Estadual de Campinas
Maria das Graças Nunes	USP/ICMC
Marilton Aguiar	UFPEL
Marley M. B. R. Vellasco	PUC-Rio
Maury Gouvêa Jr.	Pontifical Catholic University of Minas Gerais
Márcia Cristina Moraes	Colorado State University (CSU)
Márcio Basgalupp	UNIFESP

Mário Benevides	Universidade Federal do Rio de Janeiro
Myriam Delgado	Federal University of Technology of Paraná
Nelson Ebecken	Universidade Federal do Rio de Janeiro
Patrícia Tedesco	Centro de Informática - UFPE
Patricia Oliveira	University of São Paulo
Patrick Moratori	Universidade Federal Fluminense
Paulo Ferreira Jr.	Universidade Federal de Pelotas
Paulo Quaresma	Universidade de Évora
Paulo Santos	Centro Universitário da FEI
Paulo Trigo	Instituto Superior de Engenharia de Lisboa
Petrucio Viana	Universidade Federal Fluminense
Rafael Bordini	PUCRS
Regivan Hugo Nunes Santiago	Universidade Federal do Rio Grande do Norte
Reinaldo Bianchi	Centro Universitário da FEI
Rejane Frozza	UNISC
Renato Tinós	USP
Ricardo Silveira	Universidade Federal de Santa Catarina
Ricardo Tanscheit	PUC-Rio
Richard Gonçalves	UNICENTRO
Robson Motta	Universidade de São Paulo
Ronaldo Prati	Universidade Federal do ABC
Rui Camacho	University of Porto
Sandra Aluísio	USP/ICMC
Sandra Sandri	INPE
Sílvio Cazella	UFCSPA

Sergio Antônio Andrade de Freitas	Universidade de Brasília
Solange Rezende	USP/ICMC
Teresa Ludermir	Universidade Federal de Pernambuco
Thiago Pardo	USP/ICMC
Tsang Ing Ren	Universidade Federal de Pernambuco
Vasco Furtado	Universidade de Fortaleza - UNIFOR
Vinicius Melo	Federal University of São Paulo
Wagner Meira Jr.	UFMG
Wamberto Vasconcelos	University of Aberdeen
Wellington Pinheiro dos Santos	Universidade Federal de Pernambuco

Revisores Adicionais

Adolfo Guimarães	Universidade de Tiradentes
Alexandre Salle	Universidade Federal do Rio Grande do Sul
Alexandre Szabo	Mackenzie University
Aline Souza	Universidade Estadual de Campinas
André Peres	IFRS - Campus Porto Alegre
André Gustavo Maletzke	Universidade Estadual do Oeste do Paraná
Anselmo Montenegro	Universidade Federal Fluminense
Antonio Parmezan	Universidade de São Paulo
Beatriz Andrade	Universidade Federal de Sergipe
Breno Piva	Universidade Federal de Sergipe
Bruno Fernandes	Universidade de Pernambuco
Byron Leite Dantas Bezerra	Universidade de Pernambuco
Carlos Barbosa	PUC-Rio
Carlos Caminha	Universidade de Fortaleza
Carlos Marchi	Universidade Federal do ABC
Carlos Morais	UFRGS/IFRS-POA
Daniel Dantas	Universidade Federal de Sergipe

Daniel Silva	Universidade do Porto
Daniela Uez	Universidade Federal de Santa Catarina
Denis Reis	Universidade de São Paulo
Didier Vega Oliveros	Universidade de São Paulo
Diego Silva	Universidade de São Paulo
Douglas Braz	Universidade Federal de Minas Gerais
Eduardo Borges	Universidade Federal do Rio Grande
Elton Perin	Universidade Federal do Mato Grosso do Sul
Ellen Souza	Universidade Federal Rural de Pernambuco
Fabrcio Olivetti de Franca	Universidade Federal do ABC
Feng Chung Wu	Universidade Estadual do Oeste do Paran
Fernando dos Santos	Universidade do Estado de Santa Catarina
Fidelis Zanetti de Castro	Instituto Federal de Educao Cincia e Tecnologia do Esprito Santo
Flvia Barros	Universidade Federal de Pernambuco
Gabriel de Oliveira Ramos	Universidade Federal do Rio Grande do Sul
George Cabral	Universidade Federal de Pernambuco
George Cavalcanti	Universidade Federal de Pernambuco
Giancarlo Lucca	Universidad Pblica de Navarra
Gil Rocha	Universidade do Porto
Guilherme Lima	PUC-Rio
Gustavo Seeman	Universidade Federal Fluminense
Isaque Lima	Universidade Federal do Rio de Janeiro
Jarley Nbrega	Universidade Federal de Pernambuco
Jnio Canuto	Universidade Estadual de Campinas
Jefferson Santos	FGV
Jerusa Marchi	Universidade Federal de Santa Catarina
Joo Carlos Gonalves	Universidade de Coimbra
Joo Paulo Pordeus	Universidade Federal do Cear
Jonnathan Carvalho	Instituto Federal de Educao, Cincia e Tecnologia Fluminense
Jos Andr de Moura Brito	ENCE/IBGE
Karla Figueiredo	Universidade do Estado do Rio de Janeiro
Kemilly Garcia	Universidade do Porto
Lariza Oliveira	Universidade de So Paulo
Leonardo Forero	PUC-Rio
Leonardo Matos	Universidade Federal de Sergipe
Lisane Brisolaria	Universidade Federal de Pelotas
Lucas Leite	Universidade Federal do Cear
Luis Alvaro de Lima Silva	Universidade Federal de Santa Maria
Maiquel Brito	Instituto Federal de Educao, Cincia e Tecnologia do Rio Grande do Sul
Marcos Cintra	Universidade Federal Rural do Semi-rido
Maria Viviane Menezes	Universidade de So Paulo
Mariana Tasca	CTSMART

Newton Spolaôr	Universidade Estadual do Oeste do Paraná
Olacir Castro Junior	Universidade Federal do Acre
Pablo Silva	Universidade Federal Fluminense
Paulo T. Guerra	Universidade Federal do Ceará
Pedro Martins	Universidade de Coimbra
Renato Corrêa	Universidade Federal de Pernambuco
Rian Pinheiro	Universidade Federal Rural de Pernambuco
Ricardo Grunitzki	Universidade Federal do Rio Grande do Sul
Ricardo Jacomin	Universidade de São Paulo
Rui Fonseca-Pinto	Polytechnic Institute of Leiria
Silvia Schiaffino	ISISTAN - Universidad Nacional del Centro de la Pcia. de Buenos Aires
Tauã Cabreira	Universidade Federal de Pelotas
Thiago Rocha	Instituto Federal de Educação, Ciência e Tecnologia do Ceará
Tiago Schmitz	Universidade do Estado de Santa Catarina
Vitor Coelho	Universidade Federal Fluminense
Vítor Machado	Universidade Federal do Rio de Janeiro
Vlândia Pinheiro	Universidade de Fortaleza
Weber Takaki	UNICAMP
Wouter Caarls	PUC-Rio

Artigos publicados

Detecção Automática de Opiniões Falsas com base no Conteúdo das Mensagens

Emerson Freitas Cardoso, Tiago Agostinho Almeida, Renato Moraes Silva 1

Uma Abordagem de Combinação Não-Linear ARIMA-SVM para Previsão de Séries Temporais

Domingos S. de O. Santos Jr., Leonardo Guimarães de Holanda, Paulo S. G. de Mattos Neto 13

Uma Nova Variante do Algoritmo do Morcego Baseada em uma Modificação no Operador de Mutação de Michalewicz

Izabele V. O. Leite, Marcos H. F. Marcone, Fábio A. P. Paiva 25

A Preference-guided Multiobjective Evolutionary Algorithm based on Decomposition

Daniel E. Souza, Fillipe Goulart, Lucas S. Batista, Felipe Campelo 37

Simulação Multiagentes: Áreas Atingidas por Terremotos com Focos de Incêndio

Ana Júlia Fontoura, Rejane Frozza, Daniela Bagatini, Gilberto Dessbesell, Tauame Aguiar Pacce 49

Applying Machine Learning to Develop an Automatic Incident Classifier: A Massive Event Case

André Accioly Lima, Renato Carlos Pinto, Leonardo Henrique
Moreira, Célia Ghedini Ralha, Edison Ishikawa 61

**Simple Interactive Image Segmentation using Label
Propagation through kNN graphs**

Fabricio A. Breve 73

**Um Levantamento do Uso de Aprendizado Profundo em Análise
de Sentimentos**

Eduardo de O. Andrade, José Viterbo, Cristina N. Vasconcelos 85

**Aumentando o poder preditivo de classificadores lineares
através de particionamento por classe**

Nahim A. Souza, Tiago A. Almeida, Tiemi C. Sakata 97

**Online Modified Differential Evolution Method Static Stability
Control of Two-Wheeled Inverted Pendulum Vehicle Case Study**

Bruno L. Pereira, Gustavo Diniz, Leonardo Sanches, José J.P.Z.S.
Tavares 109

**Novas Funcionalidade do SIMULA - Ambiente para Simulação
em Sistemas Multiagentes Reativos**

Bruno Pick, Rejane Frozza, Daniela Bagatini, Gilberto Dessbesell 121

**Formação do Consenso por meio de Diálogos Argumentativos
em Sistemas Multiagentes**

Ayslan T. Possebom, Mariela Morveli-Espinoza, Everson Joay,
Cesar A. Tacla 133

Uma abordagem baseada em Simulated Annealing aplicada ao problema de programação de jogos do Campeonato Mineiro de Futebol

Keline M. Balieiro, Laércio I. Santos, Luciana B. Cosme, Renato A. C. Silva 145

Toward a Light Version of Convolutional Neural Network for Image Classification in Embedded Systems

Cecília F. Silva, Clauriton A. Siebra 157

Save the Data! An Intelligent Approach to Avoid Data Loss

Marcos Iseki, Bruno Nogueira, Brivaldo Junior 169

Análise de Agrupamentos através da Otimização por Busc em Grupo Melhorada por Poda

Luciano D. S. Pacífico 181

A Comparative Analysis of Neural Networks Implemented with Reduced Numerical Precision and Approximations

Vitor Ferreira Torres, Frank Sill Torres 193

Aprendizagem de máquina para predição de desempenho de estudantes de graduação na UFPE

José O. S. Brandão, Adenilton J. Silva, Roberta M. M. Gouveia, Rodrigo G. F. Soares 205

Improving the Performance of Taxi Service Applications using Multi-Agent Systems Techniques

Leonardo Blanger, Valmir Junior, Clair J. Jevinski, Alison R. Panisson, Rafael H. Bordini 217

Um Algoritmo para Agrupamento de Dados Usando Interações entre Agentes

Lutiele M. Godois, Luciano C. Marco, Diana F. Adamatti, Leonardo R. Emmendorfer 229

Analysis of unsupervised aspect term identification methods for Portuguese reviews

Mateus Tarcinalli Machado, Thiago Alexandre Salgueiro Pardo, Evandro Eduardo Seron Ruiz 239

Diagnóstico de benignidade e malignidade em imagens mamográficas utilizando Aprendizado Profundo para extração das regiões de interesse e Índices de Diversidade Filogenéticas para classificação

Ismael de H. Leal, kelson Rômulo T. Aires, Rodrigo de M. S. Veras 251

Axiom Acquisition in STRIPS Planning Domains

Romulo O. Leite, Volmir E. Wilhelm 263

DependentIE: An Open Information Extraction system on Portuguese by a Dependence Analysis

Leandro Souza de Oliveira, Rafael Glauber, Daniela Barreiro Claro 273

Rumo à integração de múltiplos QAs de domínio restrito: sistema de classificação de domínio através das perguntas

Leandro Luciani Tavares, Tiago Agostinho Almeida, Renato Moraes Silva 285

Bayesian Networks and Massey's Method applied to Soccer Outcome Prediction

Júlio César Teixeira de Oliveira, Edimilson Batista dos Santos 297

Classification of Microcalcifications in Mammographic Exams: optimization with the use of hybrid Artificial Neural Network-Genetic Algorithm (GA-ANN)

Dario Brito Calçada, Cláudio Fabiano Motta Toledo, Pedro Victor Carvalho das Chagas, Cássio Costa Almeida 309

A quantitative comparison of exhaustive discriminative pattern mining algorithms

Maurício A. S. dos Santos, Renato Vimieiro 321

Protótipo de um Método de Classificação por Descrição Mínima

Breno Lima de Freitas, Tiago A. Almeida, Renato M. Silva 333

Denoising Auto-Encoders as Feature Extractors in Hierarchical Classification Problems

Felipe Kenji Nakano, Ricardo Cerri 345

Extração de tópicos de interesse de pesquisas brasileiras em Inteligência Artificial

Marcelo K. Albertini, Adriano L. de Sá 357

Métodos de Otimização Hiperparamétrica: uma Análise Comparativa na Abordagem de Problemas de Classificação Binária Utilizando Floresta Aleatória

Wagner J. Alvarenga Jr., André P. Lemos 369

A Study of Neural Networks Models applied to Natural Language Inference

Victor G. Noronha, João C. P. da Silva 381

Evaluating Rhetorical Arguments through Fuzzy Systems

Helio H. L. C. Monte-Alto, Mariela Morveli-Espinoza, Cesar A. Tacla 393

Bayesian Sets for Binary Item Recommendation Systems

Diorge Brognara, Estevam R. Hruschka Jr. 405

Aplicando a Mineração de Opiniões na Avaliação de Relatórios de Pesquisas Científicas

João Pedro Santos Rodrigues, Zacarias Curi, Mariza Miola Dosciatti, Emerson Cabrera Paraiso 415

Improved User Cold-Start Recommendation via Two-Level Bandit Algorithms

Otávio Augusto Rodrigues, Anísio Lacerda, Flávio Luis Cardeal Pádua 427

An Architectural Approach for Developing Intelligent Personal Assistants Supported by NELL

Luís Felipe F. C. Tomazini, Oscar J. Romero, Estevam R. Hruschka Jr. 439

Towards Crime Rate Prediction through Street-level Images and Siamese Convolutional Neural Networks

Virginia Ortiz Andersson, Marco Antônio Ferreira Birck, Ricardo Matsumura Araújo and Cristian Cechinel 451

Uma abordagem Fuzzy-Adaptativa para Segmentação de Imagens de Cromossomos

Maria F. S. Andrade, Filipe R. Cordeiro, Valmir M. Filho, Fabiana F. Lima, Suy F. Hwang, Julyanne C. G. Mendonça 463

Boosting machine learning techniques for wood quality classification by particle swarm optimization

Mateus Roder, Carlos de Oliveira Affonso, André Luis Debiaso Rossi 475

Using Intelligent Agents to Optimize an Investment Portfolio

Lucas O. Souza, Celia G. Ralha 487

Classificação de Gêneros de Música Brasileira usando Letras e Acordes

Dimas Tadeu de Oliveira Junior, Tiago Fernandes Tavares 499

Multi-Task Reinforcement Learning: An Hybrid A3C Domain Approach

Marco A. F. Birck, Ulisses B. Correa, Pedro Ballester, Virginia O. Andersson, Ricardo M. Araujo 511

Detecção de Cardiopatias por Eletrocardiograma Utilizando Multilayer Perceptron

Vinícius Suterio, Cristiano Marcos Agulhari, Danilo Hernane Spatti 519

CNNFusion-Gender: metodologia de fusão de partes faciais

para a classificação de gênero em ambientes não controlados	
Flávio Henrique da Silva	531
Segmentação das fases do gesto com máquinas de vetores suporte: o impacto da subjetividade e de diferentes rotulações	
Jallysson M. Rocha, Ricardo A. Feitosa, Clodoaldo A. M. Lima, Sarajane M. Peres	543
Concept Lattice Reduction using Fuzzy Data Clustering	
Suzane Carol de Lima, Heloisa de Arruda Camargo	555
Evolução Diferencial e Busca local para a Predição da Estrutura de Proteínas	
Lucas C. Siqueira, Richard A. Gonçalves, Carolina P. Almeida, Myriam R. Delgado, Sandra M. Venske	567
Semantic Parsing Natural Language into SPARQL: an LSTM Encoder-Decoder Neural Net Approach	
Fabiano Ferreira Luz , Marcelo Finger	579
Uma Hiper-Heurística para o Flowshop Multiobjetivo	
Geovani Ferreira Antunes, Richard Aderbal Gonçalves, Sandra Mara Venske, Carolina Paula de Almeida	591
Um Estudo Experimental da Aplicação do Algoritmo de Neuroevolução com Crescimento Topológico em Imagens Médicas	
Luiz D.R. França, Péricles B.C. Miranda, Filipe R. Cordeiro	603

Seleção Adaptativa de Operador da Evolução Diferencial para o Problema da Predição Terciária da Estrutura de Proteínas

Isaque Silverio, Richard A. Gonçalves, Carolina P. Almeida, Myriam R. Delgado, Sandra M. Venske 613

A Multimodal Recommendation Framework based on Low-dimensional Latent Factors

Anna Claudia Almeida, Anisio Lacerda, Gisele Lobo Pappa, Flávio Luis Cardeal Pádua 625

Aplicação de Redes Neurais Convolucionais na Identificação de Emoções em Textos

Zacarias Curi, Alceu de Souza Britto Jr, Emerson Cabrera Paraiso 637

MuNoCC: Multiple Norms Conflict Checker for Multi-Agent Systems

Eduardo Augusto Silvestre, Viviane Torres da Silva 649

Otimização de percurso para Plotter Vertical utilizando Algoritmos Genéticos

Caio M. dos S. Escobar, Sandro Carvalho Izidoro 661

Modelagem de Funções de Transferência de Plantas Industriais em Malha Aberta e Fechada utilizando Algoritmos Genéticos

Florisvaldo C. Bomfim Jr, Igor S. Peretta, Keiji Yamanaka 673

Classificação de tweets por relevância para concepção de um modelo de aprendizado de máquina para uso em aplicações de TV Social

Cássio Faria da Silva, Cesar Augusto Camillo Teixeira, Helena de Medeiros Caseli 685

Detecção de mácula em imagens de fundo de olho por meio da metaheurística Firefly

José O. Lins, Rodrigo Veras, Leonardo Sousa 697

Estudo Sobre o Impacto do Particionamento dos Dados na Seleção Dinâmica de Atributos

Jhoseph K. L. Jesus, Anne M. P. Canuto, Daniel S. A. Araújo 709

Método Supervisionado para Categorização Automática de Lançamentos Financeiros

Cleviton V. F. Monteiro, Antonio Jorge Delgado, Rinaldo Lima, Edvan Soares 721

Evolução de Ontologias em Saúde: Um Estudo de Caso no Domínio da Doença Renal Crônica

Ana Patrícia de Sousa, Carlos Alberto de Souza, Cecília N. A. P. Gomes, Natasha Q. C. Lino 733

PACTL - SYM : um planejador baseado em Verificação Simbólica de Modelos

Viviane Bonadia dos Santos, Leliane Nunes de Barros 745

Treinamento Otimizado de Redes Neurais Artificiais Para a Classificação de Movimentos Musculares

Fábio K.H. de Barros, Aron A. Martins L., Fabio Augusto Guidotti dos Santos, Victor Hideki Yoshizumi, Rafael Martineli de Araujo, Maria Eugênia Dajer, Danilo Hernane Spatti 757

Criptografia de Fluxo por Autômatos Celulares: uma abordagem baseada na combinação de Busca Tabu com Simulated Annealing

André de Souza Brito, Stênio Sã Rosário Furtado Soares, Saulo Moraes Villela 769

Predição de Desempenho de Alunos no Ensino a Distância via Mineração de Processos

Fabio Q. Tanaka, Gabriel Lucas C. da Silva, Sarajane M. Peres, Marcelo Fantinato 781

Um pipeline para predição de genes em dados metatranscriptômicos

Alessandra P. Alves de Oliveira, Ronnie C. O. Alves 793

Encoding the Consistency of Relational Bayesian Networks

Glauber De Bona, Fabio G. Cozman 805

Aprendizado Profundo Aplicado na Classificação de Microalgas

Iago Corrêa, Paulo Drews-Jr, Márcio S. de Souza, Virgínia Maria Tavano 817

Application of Bicluster Algorithms in Human Aging Database

Marta Dias Moreira Noronha, Caio Eduardo Ribeiro, Luis Enrique Zárate 827

Planning as Mixed-Horn Formulas Satisfiability

Razer A. N. R. Montaña, Bruno C. Ribas 839

Uma Hiper-Heurística para o Problema Quadrático de Alocação Multiobjetivo

Geovani Ferreira Antunes, Richard Aderbal Gonçalves, Sandra Mara Venske, Carolina Paula de Almeida 851

Redes Neurais Convolucionais Aplicadas ao Processo de Classificação de Cultivares de Guaranazeiros

Allex de Lima Sousa, Marcos Filipe Alves Salame, Firmino José do Nascimento Filho, André Luiz Atroch 863

Combinando novidade e popularidade em recomendações personalizadas através do aprendizado do perfil do usuário

Ricardo M. Bertani, Reinaldo A. C. Bianchi, Anna H. R. Costa 873

Um algoritmo GVNS usando o framework KHE para resolver problemas de programação de horários em escolas

Ulisses Rezende Teixeira, Marccone Jamilson Freitas Souza, Sérgio Ricardo de Souza 885

**Classificação de Emoções Discretas em Sinais de EEG
Utilizando Técnicas de Aprendizado de Máquina**

Vancley Oliveira Simão, Heitor Scalco Neto, Wilian Soares Lacerda 897

**Estimation of Origin-Destination Matrix from Traffic Counts in
the City of Porto with PSO and Taxi Trip Data**

Luciano Urgal Pando, Ricardo Lüders 909

**Transferência de aprendizagem via método de regressão por
vetores suporte**

Rodolfo S. Simões, Patrícia R. Oliveira, Clodoado A. M. Lima 921

**Sistema Fuzzy Genético Baseado em Regras: Um Operador de
Pré-Seleção com Foco na Diversidade das Soluções**

Eduardo Fernando Velludo Prado, Maykon Rocha Santana, Tiago
Pinho da Silva, Marcela Aparecida Aniceto dos Santos, Heloisa de
Arruda Camargo 933

**Metodologia automática para diagnóstico do glaucoma usando
abordagem Deep Learning**

Marcos Vinícius dos Santos Ferreira, Alcilene D. De Sousa, Antônio
Oseas de C. Filho, Patrícia Medyna L. de L. Drumond, Patrícia
Vieira da S. Barros 945

**Impact of Base Partition Selection on Multi-Objective Clustering
Ensemble**

Gabriel Leonardo Pedote, Katti Faceli, Tiemi Christine Sakata 957

**Análise de dados obtidos através de um sistema de telemetria
automotivo utilizando K-NN**

Wilson Borba da Rocha Neto, José Maria Pires de Menezes Jr.,
Rhulio Victor Luz Souza 969

D2DSS : Simulador Discreto 2D do Jogo de Futebol

Igor Oliveira Borges, Valdinei Freire, Karina Valdivia Delgado

981

Detecção Automática de Opiniões Falsas com base no Conteúdo das Mensagens

Emerson Freitas Cardoso¹, Tiago Agostinho Almeida¹, Renato Moraes Silva¹

¹Departamento de Computação (DComp)
Universidade Federal de São Carlos (UFSCar) – Sorocaba – SP – Brasil

mersu898@gmail.com, talmeida@ufscar.br,
renatoms@dt.fee.unicamp.br

Abstract. *Some websites facilitate the sharing of online reviews and, at the same time, aim to attract the confidence of the users. However, the increasing competition between service providers and product manufacturers are resulting in a growing volume of spam comments. In this case, people are hired to post fake reviews with an attempt to promote or defame products or services. Such messages can cause direct and indirect losses to businesses and users. In addition, they are created to appear authentic and, therefore, their detection by humans is very difficult. In this scenario, this paper aims to answer important still opened questions and moreover it offers and evaluates a proposal to automatically filter these messages using the MDLText, a robust, fast, multi-class, and naturally incremental text categorization approach. The experiments performed, along with the statistical analysis of the results, demonstrated this solution is promising and can be successfully applied in practice.*

Resumo. *Alguns sites oferecem recursos para facilitar o compartilhamento de reviews online e, ao mesmo tempo, atrair a confiança dos usuários nesses reviews. No entanto, a competição entre prestadores de serviços e fabricantes vêm fomentando casos de spam reviews: pessoas são contratadas para publicar reviews falsos com o objetivo de promover ou difamar produtos ou serviços, o que pode causar prejuízos diretos e indiretos para empresas e usuários. Spam reviews são criados para parecerem autênticos e, portanto, apresentam elevada dificuldade para serem detectados por seres humanos. Neste cenário, este artigo oferece respostas para importantes perguntas ainda em aberto e avalia a proposta de filtrar automaticamente esses reviews usando o MDLText, um método de classificação robusto, rápido, multiclasse e incremental. Os experimentos realizados, juntamente com a análise estatística dos resultados, comprovam que a solução apresentada é promissora e pode ser utilizada com sucesso na prática.*

1. Introdução

Hoje em dia, diversos sites, como o TripAdvisor, Yelp e Foursquare, hospedam e compartilham opiniões (*reviews*) de usuários sobre serviços, produtos e lugares. Praticamente todos os grandes sites de *e-commerce*, como a Amazon, também oferecem recursos para compartilhar *reviews* sobre os produtos comercializados. Essa facilidade faz com que as pessoas confiem cada vez mais nos *reviews online* e aumenta a competitividade entre fabricantes de produtos e donos de estabelecimentos comerciais, que procuram ganhar visibilidade nessas mídias. Contudo, diante deste cenário, começaram a surgir casos de opiniões falsas, geralmente publicadas por usuários contratados pelas empresas com o objetivo de difamar ou promover produtos ou serviços. Em um caso recente, a Samsung

foi processada por pagar pessoas para escreverem opiniões falsas usadas para difamar a empresa concorrente HTC e, ao mesmo tempo, promover produtos da própria marca¹.

Opiniões falsas são denominadas *spam* de opinião, *spam reviews*, *fake reviews*, *deceptive reviews* ou *fraudulent reviews*. Os autores desse tipo de conteúdo são denominados *spammers*. *Spam reviews* podem causar grande prejuízo para fabricantes de produtos ou provedores de serviços, que muitas vezes são prejudicados com opiniões falsas negativas que desvalorizam sua marca e seus produtos, ao mesmo tempo em que promovem rapidamente produtos dos seus concorrentes.

A disseminação desse tipo de conteúdo é um problema preocupante, pois ele representa uma porcentagem expressiva dos *reviews online*. Segundo [Luca e Zervas 2015], cerca de 16% dos *reviews* sobre restaurantes do Yelp são *spam*. Pesquisas² também apontam que, até 2014, entre 10% e 15% dos *reviews* nas mídias sociais eram falsos. Para diminuir a propagação dessas mensagens, a Amazon recentemente adotou uma funcionalidade denominada AVP (*Amazon Verified Purchase*) que, basicamente, indica se o autor de um *review* efetuou a compra do produto. Ainda, ela tem bloqueado e tomado medidas judiciais contra vendedores que postam *spam reviews* para promover suas mercadorias³.

Devido ao crescente aumento no volume de opiniões falsas, as redes sociais passaram a disponibilizar mecanismos para que usuários denunciem publicações que possivelmente sejam *spam*. No entanto, de acordo com [Harris 2012], é muito difícil que seres humanos consigam detectar *spam reviews* com precisão, pois eles são cuidadosamente criados para parecerem autênticos. Para ilustrar essa dificuldade, abaixo são apresentados dois exemplos reais, sendo que um deles é legítimo e o outro é *spam* [Jindal e Liu 2007].

I have stayed at many hotels traveling for both business and pleasure and I can honestly say that The James is tops. The service at the hotel is first class. The rooms are modern and very comfortable. The location is perfect within walking distance to all of the great sights and restaurants. Highly recommend to both business travellers and couples.

My husband and I stayed at the James Chicago Hotel for our anniversary. This place is fantastic! We knew as soon as we arrived we made the right choice! The rooms are BEAUTIFUL and the staff very attentive and wonderful!! The area of the hotel is great, since I love to shop I couldn't ask for more!! We will definitely be back to Chicago and we will for sure be back to the James Chicago.

Neste cenário, é crescente a necessidade de empregar recursos automáticos e eficientes para filtrar esses tipos de mensagens. Sendo assim, este artigo oferece uma análise do desempenho do MDLText, um novo método de categorização de textos baseado no princípio da descrição mais simples (MDL – *Minimum Description Length*) [Silva et al. 2017b, Silva et al. 2017a], e o compara com métodos tradicionais de classificação. Além disso, este trabalho procura oferecer respostas para as perguntas ainda em aberto, mostradas a seguir.

P1: O MDLText pode oferecer desempenho estatisticamente superior aos métodos tradicionais de classificação?

¹ABC News: *Samsung Fined For Paying People to Criticize HTC's Products*. Disponível em: <https://goo.gl/hKsCQW>. Acessado em: 16/05/2017.

²Gartner Newsroom: *Gartner Says By 2014, 10-15 Percent of Social Media Reviews to Be Fake, Paid for By Companies*. Disponível em: <http://www.gartner.com/newsroom/id/2161315>. Acessado em: 16/05/2017.

³TechCrunch: *Amazon sues more sellers for buying fake reviews*. Disponível em: <https://goo.gl/Ctt00L>. Acessado em: 16/05/2017.

- P2:** O desempenho dos métodos pode ser afetado pela polaridade das opiniões (reclamações × elogios)?
- P3:** O desempenho dos métodos é preservado ao classificar amostras reais de *spam*?
- P4:** O desempenho dos métodos pode ser afetado pela diversidade de objetos de análise (treinamento com opiniões de múltiplos tipos de serviço/mercadoria × um único tipo de serviço/mercadoria)?

O restante deste trabalho está estruturado da seguinte forma: na Seção 2, são apresentados alguns trabalhos correlatos. Na Seção 3, o método MDLText é detalhado. Na Seção 4, são apresentadas as bases de dados e as configurações adotadas na avaliação experimental. Os resultados obtidos são reportados na Seção 5. Na Seção 6, são apresentadas as principais conclusões e direcionamentos para trabalhos futuros.

2. Contextualização bibliográfica

Os estudos de [Jindal e Liu 2007] foram pioneiros na área de detecção de *spam reviews*. Eles ofereceram uma base de dados composta por milhões de *reviews* coletados da Amazon. Segundo os autores, os *spams* encontrados podem ser divididos em três categorias:

1. *opiniões falsas: reviews* enganosos que promovem ou difamam produtos;
2. *opiniões sobre marcas: reviews* vinculados a um produto, mas que analisam somente a marca ou empresa que o fabrica;
3. *mensagens sem opinião: reviews* que contém propagandas ou textos não relacionados com o tema ou produto.

De acordo com [Jindal e Liu 2007], os tipos (2) e (3) são mais fáceis de serem detectados através de métodos tradicionais de aprendizado de máquina. Contudo, o tipo (1) é mais complexo de ser identificado até mesmo por humanos, o que, conseqüentemente, inviabiliza a criação de bases de dados rotuladas necessárias para treinar os métodos de classificação. Ainda, segundo os autores, técnicas empregadas na detecção de *web spam* ou *email spam* não são adequadas para detectar opiniões falsas. Para contribuir com a solução do problema, eles aplicaram um método para encontrar *reviews* duplicados ou aproximadamente duplicados. Depois, eles geraram uma base de dados, em que os *reviews* duplicados foram rotulados como *spam* e os demais como não-*spam* (legítimos).

[Jindal e Liu 2008] avaliaram o desempenho do método de regressão logística na detecção de opiniões falsas. Segundo os autores, o método foi bastante efetivo, mas novas abordagens ainda precisariam ser estudadas para aumentar a eficácia da predição em *spam reviews* sofisticados, como aqueles escritos cuidadosamente para parecerem autênticos (não sendo necessariamente duplicados de outros *reviews*).

[Ott et al. 2011] testaram atributos de psicolinguística representados por unigramas, bigramas e trigramas para detectar sentenças enganosas no texto. A base de dados utilizada é composta por *reviews* legítimos e positivos sobre um conjunto de hotéis do TripAdvisor e os *spam reviews* foram criados manualmente, através da contratação de serviços pelo site *Amazon Mechanical Turk*⁴. Os autores avaliaram os desempenhos dos métodos *Naïve Bayes* e *SVM*, sendo que a melhor acurácia foi de aproximadamente 89%.

O trabalho apresentado por [Ott et al. 2011] foi estendido por [Ott et al. 2013], que analisaram bases de *reviews* com polaridade negativa (reclamações). Com isso, os autores publicaram a primeira base de dados *gold-standard* de *spam reviews* com polaridade negativa. Os *spam reviews* que a compõe foram escritos por pessoas contratadas através do site *Amazon Mechanical Turk* e as opiniões legítimas foram extraídas de diversos sites, tais como o TripAdvisor, Hotels.com e Yelp. Segundo os autores, o melhor resultado obtido ofereceu uma acurácia de aproximadamente 86%.

⁴Amazon Mechanical Turk. Disponível em <https://mturk.com>. Acessado em: 15/05/2017.

Outro trabalho que merece destaque é o de [Harris 2012], onde ele utilizou atributos contendo métricas sobre a complexidade dos textos dos *reviews*, como por exemplo a medida ARI (*Automated Readability Index*). Primeiramente, foram realizados testes observando a taxa de acerto da detecção por humanos utilizando apenas o conteúdo dos *reviews*. Posteriormente, foram oferecidas as métricas de complexidade dos textos, com o objetivo de auxiliar a detecção. A acurácia obtida pelo método SVM foi de aproximadamente 76%, enquanto que a acurácia obtida pelos humanos foi de aproximadamente 58% no primeiro teste e 65% no segundo teste. Uma das conclusões obtidas pelos autores é a de que os seres humanos detectam *hyper spam* (opiniões que visam promover algo) mais facilmente do que *defaming spam* (opiniões que visam difamar algo).

[KC e Mukherjee 2016] realizaram uma análise sobre os *spam reviews* de restaurantes do Yelp, com o objetivo de encontrar padrões temporais do comportamento dos *spammers*. Os autores utilizaram séries temporais sobre as postagens de *reviews* genuínos e enganosos. Após a análise, foi descoberta uma forte correlação entre os comportamentos de postagens de *spam* e as séries temporais de postagens de *reviews* genuínos.

Um dos problemas na classificação de *spam reviews* é o desbalanceamento das classes. Para lidar com isso, [Al Najada e Zhu 2014] propuseram um método que divide a base de *spam reviews* desbalanceada em múltiplas bases de dados, depois aplica subamostragem aleatória para balancear as classes e, posteriormente, gera um classificador para cada sub-base de dados. Assim, quando uma amostra não-rotulada é apresentada, ela recebe a classe retornada pela maioria dos classificadores. Segundo os autores, o método proposto obteve um desempenho superior ao método C4.5.

[Crawford et al. 2015] analisaram os trabalhos da literatura que abordam o problema de detecção de *spam reviews*, considerando estudos que envolvem métodos supervisionados, não-supervisionados e semi-supervisionados. Eles constataram que os métodos mais usados são os de aprendizado supervisionado utilizando o esquema *bag of words* para a representação do conteúdo dos *reviews*. Além disso, eles verificaram que o SVM é o que obteve o melhor desempenho na maioria dos estudos. Ainda, foi observado que diversos trabalhos empregam bases de dados com *spam reviews* artificiais, devido a dificuldade em se obter bases de dados com amostras reais rotuladas adequadamente.

Apesar dos estudos já realizados, ainda não há um consenso sobre quais seriam as respostas para as perguntas P1 a P4, apresentadas no final da Seção 1. Sendo assim, o objetivo principal deste trabalho é (i) analisar e comparar o desempenho do MDLText, um método de categorização de texto publicado recentemente e que vem obtendo desempenho elevado em diversos problemas de classificação de texto [Silva et al. 2017b, Silva et al. 2017a] e (ii) oferecer respostas apropriadas para cada um dos questionamentos levantados, através de criteriosa avaliação experimental.

3. MDLText

O MDLText é um método de categorização de texto que suporta múltiplas classes e aprendizado incremental (*online*) [Silva et al. 2017b]. Ele é baseado no princípio MDL, cuja ideia principal é que em um problema onde é necessário escolher entre dois ou mais modelos para se ajustar a um determinado conjunto de dados, deve-se escolher aquele que possui o menor tamanho de descrição [Rissanen 1978, Rissanen 1983]. Isso significa que modelos menos complexos são preferíveis [Barron et al. 1998, Grünwald et al. 2005]. Assim, o princípio MDL é visto como uma formalização do princípio da parcimônia, também conhecido como Navalha de Occam.

Os modelos avaliados pelo MDLText representam as possíveis classes do problema. Quando um documento de texto d não rotulado é apresentado, a ele é atribuída a

classe que possui o menor tamanho de descrição. Para isso, ele utiliza a seguinte equação:

$$c(d) = \operatorname{argmin}_{\forall c} L(d|c_j). \quad (1)$$

$L(d|c_j)$ corresponde ao tamanho de descrição do documento d que está sendo classificado, quando é representado pelo modelo relativo à classe c_j . $L(d|c_j)$ é calculado pela seguinte equação:

$$L(d|c_j) = \left[\sum_{i=1}^{|d|} L(t_i|c_j) \times K(t_i) \right] \times \hat{S}(d, c_j), \quad (2)$$

sendo que:

- $L(t_i|c_j)$ é o tamanho de descrição de cada termo quando representado pelo modelo relativo a cada classe;
- $K(t_i)$ é uma penalidade para cada termo do documento baseado em quanto ele pode ser relevante para a identificação da classe;
- $\hat{S}(d, c_j)$ corresponde a uma penalidade baseada na similaridade entre o documento que está sendo classificado e um protótipo da classe.

O tamanho da descrição de cada termo ($L(t_i|c_j)$) é calculado pela Equação 3 e é baseado na codificação de Shannon-Fano, conforme explicado em [Silva et al. 2017b].

$$L(t_i|c_j) = \lceil -\log_2 \beta(t_i|c_j) \rceil. \quad (3)$$

$\beta(t_i|c_j)$ é um peso condicional associado a cada termo, relativo a todos os outros termos que já apareceram em documentos da classe. Ele é calculado por:

$$\beta(t_i|c_j) = \frac{W(t_i|c_j) + \frac{1}{|\Omega|}}{W(c_j) + 1}. \quad (4)$$

A função $W(t_i|c_j)$ representa a soma de pesos TF-IDF normalizados do termo t_i , para os documentos da classe c_j . A função $W(c_j)$ representa a soma dos pesos TF-IDF normalizados para todos os termos de todos os documentos da classe c_j . O parâmetro Ω é utilizado para preservar uma porção do tamanho de descrição para termos desconhecidos, não observados durante o treinamento. Segundo [Silva et al. 2017b], o melhor valor para esse parâmetro pode variar para cada problema e, portanto, uma busca em grade pode ser utilizada para encontrar o valor mais adequado.

Na Equação 2, a função $K(t_i)$ representa uma penalidade para um determinado termo e seu objetivo é aumentar a capacidade de separação de classes. A Equação 5 apresenta como ela é calculada.

$$K(t_i) = \frac{1}{(1 + \alpha) - F(t_i)}. \quad (5)$$

O parâmetro α é usado para evitar um denominador igual a zero. A função $F(t_i)$ retorna um valor entre 0 e 1 que corresponde à importância do termo t_i na separação das classes. Segundo [Silva et al. 2017b], para calcular esse valor pode ser usada a técnica de fatores de confiança (CF – *Confidence Factors*) proposta por [Assis et al. 2006].

Na Equação 2, a função $\hat{S}(d, c_j)$ representa uma penalidade aplicada de acordo com a dissimilaridade entre d e um vetor protótipo da classe c_j . Quanto menor a similaridade entre os dois, maior será essa penalidade. Para isso, é usada a seguinte equação:

$$\hat{S}(d, c_j) = -\log_2 \left(\frac{1}{2} \times \frac{\sum_{i=1}^{|d|} \hat{w}(t_i, d|c_j) \times \bar{c}_j(t_i)}{\|\hat{w}(:, d)\|_2 \times \|\bar{c}_j\|_2} \right), \quad (6)$$

onde $\hat{w}(:, d)$ é a soma dos pesos TF-IDF normalizados dos termos presentes em d e \bar{c}_j é o vetor protótipo, que é formado pela média dos pesos dos termos que ocorreram nos documentos de treinamento da classe c_j .

A Equação 1, usada para definir a classe de um documento não-rotulado, e todas as demais equações apresentadas são baseadas apenas na frequência dos termos e nos pesos TF-IDF. Portanto, o estágio de treinamento do MDL_{Text} consiste apenas na coleta dessas informações a partir dos documentos de treinamento. Consequentemente, esse processo pode ser feito de maneira incremental, o que torna esse método escalável e apto a ser empregado em cenários de classificação *online*, onde novos exemplos podem ser usados para atualizar o modelo de predição previamente treinado. Além disso, tanto o estágio de treinamento quanto de classificação possuem complexidade computacional linear [Silva et al. 2017b]. Essa característica é muito desejável em aplicações reais de classificação, pois geralmente envolvem um grande volume de dados.

4. Metodologia experimental

Para a reprodutibilidade desta pesquisa, nesta seção, são detalhadas as etapas da avaliação experimental conduzida para avaliar o desempenho dos métodos de aprendizado de máquina aplicados na filtragem de opiniões falsas com base no conteúdo das mensagens.

4.1. Bases de dados

Para dar credibilidade aos resultados, foram utilizadas duas grandes coleções de dados bem conhecidas, reais e públicas:

- *TripAdvisor* [Ott et al. 2011]: possui *reviews* legítimos de hotéis de Chicago. Os *spam reviews* não são reais e foram escritos por pessoas contratadas *online* pelos autores, através do site *Amazon Mechanical Turk*. A única informação disponibilizada é o conteúdo textual dos *reviews* e sua respectiva classe (*spam* ou legítima).
- *Yelp* [Mukherjee et al. 2013]: possui *reviews* de hotéis e restaurantes dos Estados Unidos. Os *reviews* rotulados como *spam* são reais e foram detectados pelo filtro proprietário do Yelp. Os demais *reviews* são legítimos. O filtro do Yelp, porém, pode gerar ruídos na classificação, uma vez que, além do conteúdo dos *reviews*, também leva em consideração meta-parâmetros, como informações de autores, onde usuários novos ou usuários que publicaram mensagens muito curtas podem apresentar um grau de suspeita que pode ser detectado pelo filtro.

Nessas coleções, foram selecionadas apenas as amostras de *reviews* escritos na língua inglesa. Os textos foram convertidos para letras minúsculas e segmentados em múltiplos termos utilizando como delimitadores quaisquer caracteres não alfanuméricos. A representação utilizada foi a tradicional *bag of words*, sem a aplicação de nenhuma técnica de pré-processamento, como lematização, estemização ou remoção de *stop words*. Essa abordagem foi escolhida como tentativa de detectar características sutis de escrita que possam ajudar na classificação correta das opiniões falsas e, portanto, ainda não foram investigados outros tipos de atributos para este trabalho. O esquema de pesos utilizado para representar os termos é detalhado na Seção 4.3.

As coleções de *reviews* foram divididas de acordo com os tipos de amostras: (i) amostras com polaridade positiva (*reviews* positivos ou elogios), (ii) negativa (*reviews* negativos ou reclamações) e (iii) com ambas as polaridades. As informações sobre cada coleção de dados são mostradas na Tabela 1. A coluna ID corresponde à sigla da coleção utilizada na descrição dos experimentos e análises subsequentes, $|D|$ corresponde ao total de amostras, $|T|$ é o tamanho do vocabulário, $\#Spam$ representa o total de amostras rotuladas como *spam*, $\#Ham$ indica o total de amostras legítimas, M é a mediana do número de termos por amostra e IQR corresponde à amplitude interquartil (*Interquartile Range*) do número de termos por amostra.

Tabela 1. Informações básicas sobre as coleções de dados.

Coleção de dados	Polaridade	ID	$ D $	$ T $	$\#Spam$	$\#Ham$	M	IQR
TripAdvisor - Hotéis	Pos	T-H-P	800	5.548	400	400	71	40
TripAdvisor - Hotéis	Neg	T-H-N	800	7.596	400	400	99	53
TripAdvisor - Hotéis	Pos + Neg	T-H-PN	1.600	9.571	800	800	84	51
Yelp - Hotéis	Pos	Y-H-P	916	7.204	500	416	71	70
Yelp - Hotéis	Neg	Y-H-N	684	7.708	400	284	98	82
Yelp - Hotéis	Pos + Neg	Y-H-PN	1.600	10.673	900	700	80	78
Yelp - Restaurantes	Pos	Y-R-P	1.200	10.358	600	600	69	86
Yelp - Restaurantes	Neg	Y-R-N	1.200	10.146	600	600	83	73
Yelp - Restaurantes	Pos + Neg	Y-R-PN	2.400	14.923	1.200	1.200	77	80
Yelp - Hotéis + Rest.	Pos	Y-HR-P	2.116	13.401	1.100	1.016	70	78
Yelp - Hotéis + Rest.	Neg	Y-HR-N	1.884	13.380	1.000	884	87	76
Yelp - Hotéis + Rest.	Pos + Neg	Y-HR-PN	4.000	19.043	2.100	1.900	78	80

É possível observar que, em geral, o número médio de termos que compõe cada amostra é muito menor que o número total de termos (tamanho do vocabulário). Isso significa que as amostras são representadas por vetores de alta dimensionalidade e bastante esparsos, o que pode afetar o desempenho dos métodos de classificação. Outro destaque é para o desbalanceamento no número de amostras por classe das coleções de *reviews* de hotéis do Yelp e de *reviews* de hotéis + restaurantes do Yelp. Essa característica também pode prejudicar o desempenho dos métodos, tornando-os tendenciosos à classe majoritária. Além disso, é interessante observar que a quantidade de termos em *reviews* com polaridade negativa é sempre maior que os *reviews* com polaridade positiva. Isso indica que, em média, os usuários tendem a escrever textos mais longos quando fazem reclamações do que quando expressam elogios.

4.2. Medida de desempenho

Foi adotada como medida de desempenho a tradicional F-medida. Basicamente, ela calcula a média harmônica entre a precisão e a revocação de um classificador e seus valores variam entre 0 (pior desempenho) e 1 (melhor desempenho).

4.3. Métodos

O MDLText foi comparado aos seguintes métodos consolidados de classificação:

- *Naïve Bayes* multinomial (M.NB) [McCallum e Nigam 1998];
- *Naïve Bayes* Bernoulli (B.NB) [McCallum e Nigam 1998];
- *K-nearest neighbours* (KNN) [Cover e Hart 1967];
- *Decision trees* (DT) [Breiman et al. 1984];
- *Random forest* (RF) [Breiman 2001];
- Rocchio [Rocchio 1971, Manning et al. 2009]; e
- *Linear support vector machines* (SVM) [Boser et al. 1992, Cortes e Vapnik 1995].

O MDLText⁵ foi avaliado por meio do programa disponibilizado por [Silva et al. 2017b], cuja implementação foi feita em linguagem C++. Todos os outros métodos e experimentos foram implementados usando a linguagem de programação Python, através de funções disponibilizadas pela biblioteca `scikit-learn`⁶.

Para o SVM, foi empregada a função de *kernel* linear, pois ela é a mais indicada para problemas com alta dimensionalidade [Hsu e Lin 2002]. Além disso, uma vez que o desempenho dos métodos SVM, KNN, RF e MDLText pode ser drasticamente afetado pelas escolhas de seus meta-parâmetros, foi realizada uma busca em grade com validação cruzada *5-fold* estratificada para encontrar os melhores valores. A Tabela 2 apresenta o intervalo de valores percorridos.

Tabela 2. Intervalo de valores analisados na busca em grade.

Método	Parâmetro	Intervalo
SVM	C (custo)	$(2^{-15}, 2^{15})$
KNN	K (número de vizinhos mais próximos)	$(5, 50)$
RF	N (número de árvores de decisão)	$(10, 150)$
MDLText	Ω (parâmetro usado na Equação 4)	$(2^1, 2^{25})$

Outro ponto importante na configuração dos experimentos é que, para a maioria dos métodos, o esquema de pesos utilizado para representação dos *reviews* foi o TF-IDF (*term frequency-inverse document frequency*), exceto para o método B.NB, onde o esquema binário de pesos foi escolhido, pois é intrínseco ao método.

5. Resultados

A Tabela 3 apresenta os resultados obtidos nos experimentos. Cada valor representa a média da F-medida obtida usando validação cruzada *5-fold* estratificada. Os valores em negrito indicam o melhor resultado para cada coleção de dados.

Tabela 3. Resultados obtidos na avaliação experimental

ID	B.NB	DT	KNN	MDLText	M.NB	RF	Rocchio	SVM
T-H-P	0,870	0,686	0,849	0,870	0,889	0,861	0,891	0,899
T-H-N	0,835	0,662	0,684	0,848	0,760	0,848	0,785	0,873
T-H-PN	0,846	0,681	0,761	0,866	0,846	0,848	0,843	0,878
Y-H-P	0,626	0,564	0,614	0,764	0,568	0,650	0,638	0,648
Y-H-N	0,613	0,529	0,599	0,701	0,588	0,604	0,610	0,607
Y-H-PN	0,616	0,557	0,609	0,750	0,566	0,637	0,615	0,633
Y-R-P	0,778	0,792	0,802	0,894	0,762	0,843	0,828	0,869
Y-R-N	0,595	0,556	0,542	0,722	0,551	0,600	0,569	0,583
Y-R-PN	0,680	0,632	0,623	0,735	0,566	0,703	0,677	0,687
Y-HR-P	0,681	0,649	0,692	0,719	0,637	0,721	0,719	0,750
Y-HR-N	0,564	0,535	0,523	0,708	0,535	0,605	0,569	0,577
Y-HR-PN	0,633	0,588	0,583	0,717	0,548	0,651	0,632	0,639

Individualmente, o SVM obteve o melhor desempenho nas coleções de dados do TripAdvisor, enquanto que o MDLText obteve os melhores resultados para as coleções do Yelp. Além disso, os resultados obtidos por todos os métodos foram inferiores quando

⁵MDLText. Disponível em <https://github.com/renatoms88/MDLText>. Acessado em 15/05/2017.

⁶Scikit learn. Disponível em <http://scikit-learn.org/stable/>. Acessado em 15/05/2017.

foram utilizados *spam reviews* reais (coleções do Yelp). Contudo, a queda de desempenho do MDLText foi menos acentuada que a dos outros métodos. Um possível motivo para esse comportamento pode estar relacionado com a característica inerente do MDLText de evitar sobreajustamento aos dados, pois o modelo mais simples é privilegiado.

O melhor resultado geral (F-medida = 0,899) foi obtido pelo SVM usando a coleção de *reviews* de hotéis do TripAdvisor com avaliações de polaridade positiva. Já, o pior resultado geral (F-medida = 0,523) foi obtido pelo KNN, para a coleção com *reviews* de hotéis e restaurantes do Yelp, com polaridade negativa.

É possível observar que os resultados obtidos com opiniões de polaridade positiva foram superiores em relação aos de avaliações negativas. Este fato indica que o problema pode ser melhor explorado separadamente, com diferentes modelos para cada polaridade. Como esperado, a junção de *reviews* positivos com negativos resultou em um desempenho intermediário entre os resultados individuais das polaridades.

Em geral, como os métodos obtiveram resultados superiores para as coleções de *spam reviews* criados artificialmente, conclui-se também que essas amostras não refletem fielmente a realidade e podem conduzir a uma visão muito otimista do desempenho dos métodos de classificação. Outra característica que pode ser observada é que os resultados com *reviews* de restaurantes do Yelp foram, em geral, superiores aos de *reviews* de hotéis. Além disso, como esperado, a junção dos *reviews* de hotéis e de restaurantes ocasionou uma leve piora em relação aos resultados obtidos usando apenas *reviews* de restaurantes. Isso indica que, provavelmente, induzir um modelo de predição para cada tipo de produto/serviço poderá oferecer melhores resultados.

Para certificar que os resultados reportados na Tabela 3 não foram obtidos ao acaso, foi realizada uma análise estatística usando o teste não paramétrico de Friedman, seguindo a metodologia descrita em [Demšar 2006]. O teste de Friedman usa o *ranking* médio de cada método avaliado para verificar se a hipótese nula, que afirma que os resultados obtidos são estatisticamente equivalentes, pode ser descartada.

Para um intervalo de confiança $\alpha = 0,05$, o teste de Friedman indicou que a hipótese nula foi rejeitada. Diante disso, para verificar se o desempenho obtido pelo MDLText foi significativamente superior aos demais métodos, foi feita uma análise estatística usando o teste *post-hoc* de Bonferroni-Dunn [Demšar 2006] (Figura 1). Se a diferença entre os *rankings* for maior do que uma diferença crítica, então conclui-se que há diferença estatística entre os resultados.

Para um intervalo de confiança $\alpha = 0,05$, a diferença crítica foi igual a 2,69. Com base nos *rankings* médios de cada método e no valor da diferença crítica, há evidência estatística de que o desempenho obtido pelo MDLText foi superior aos métodos DT, KNN, B.NB e M.NB. Apesar do *ranking* médio do MDLText ter sido melhor do que o dos métodos SVM, RF e Rocchio, a diferença entre eles ficou abaixo da diferença crítica e, portanto, eles foram considerados estatisticamente equivalentes.

6. Conclusões

Reviews online são compartilhados e visualizados diariamente por milhões de usuários da Internet e beneficiam desde consumidores em potencial até donos de negócios e fabricantes de produtos. Eles exercem cada vez mais influência em tomadas de decisões e, por isso, vêm despertando o interesse de pessoas mal-intencionadas que propagam *spam reviews*. Normalmente, essas opiniões falsas são escritas para enganar consumidores, com o objetivo de aumentar a popularidade de um produto ou afetar negativamente sua imagem. Esta prática fraudulenta pode causar prejuízos para consumidores, fabricantes de produtos, donos de estabelecimentos e *sites* especializados em *reviews*.

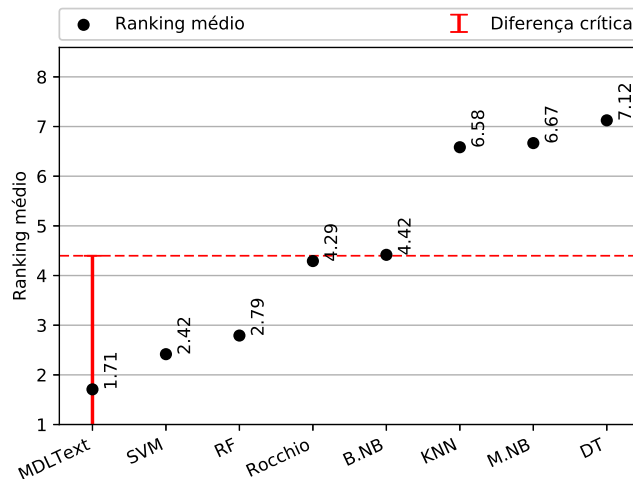


Figura 1. *Rankings* médios usados na análise estatística e diferença crítica calculada usando o teste *post-hoc* de Bonferroni–Dunn

Devido ao grande volume de opiniões compartilhadas diariamente e ao crescente aumento na quantidade de opiniões falsas, faz-se necessário o emprego de filtros para detectar e bloquear automaticamente essas mensagens, a fim de mitigar esse problema. Nesse cenário, este trabalho avaliou o desempenho do MDLText, um novo método de categorização de textos baseado no princípio da descrição mais simples. A escolha desse método foi influenciada pelas suas características que são desejáveis na aplicação prática para a solução do problema, tais como: (1) alta velocidade, pois possui complexidade computacional linear; (2) escalabilidade, pois o treinamento pode ser realizado de forma incremental; e (3) robustez ao problema de sobreajustamento aos dados.

Revisitando as perguntas destacadas no final da Seção 1, as respostas encontradas foram as seguintes.

P1: *O MDLText pode oferecer desempenho estatisticamente superior aos métodos tradicionais de classificação?*

Experimentos foram realizados com duas coleções de dados e o desempenho do MDLText foi comparado aos seguintes métodos tradicionais de classificação de texto: SVM, RF, Rocchio, B.NB, KNN, M.NB e DT. Neste caso, o SVM obteve os melhores resultados para as bases do TripAdvisor, enquanto que o MDLText obteve os melhores resultados para amostras do Yelp. No geral, o MDLText obteve o melhor *ranking* médio e a análise estatística dos resultados evidenciou que o desempenho dele foi superior ao dos métodos DT, KNN, B.NB e M.NB e estatisticamente equivalente aos resultados do SVM, RF e Rocchio.

P2: *O desempenho dos métodos pode ser afetado pela polaridade das opiniões (reclamações × elogios)?*

Foi observada uma queda de desempenho nos experimentos que envolveram classes desbalanceadas e amostras com polaridade negativa (reclamações ou opiniões difamatórias). Assim, ficou evidente que o emprego de diferentes modelos de predição treinados com amostras separadas por polaridade e com quantidade balanceada de exemplos pode trazer ganho de desempenho.

P3: *O desempenho dos métodos é preservado ao classificar amostras reais de spam?*

Os resultados evidenciaram que todos os métodos obtiveram melhor desempenho com bases de *spam reviews* criadas artificialmente. Foi observada uma queda geral de desempenho quando amostras reais de *spam reviews* foram utilizadas. Portanto,

conclui-se que utilizar amostras de *spam* criadas artificialmente pode conduzir a uma visão muito otimista do desempenho dos métodos.

P4: *O desempenho dos métodos pode ser afetado pela diversidade de objetos de análise (treinamento com opiniões de múltiplos tipos de serviço/mercadoria × um único tipo de serviço/mercadoria)?*

Os resultados experimentais indicaram que *reviews* de hotéis do Yelp foram mais difíceis de classificar do que os *reviews* de restaurantes. Consequentemente, é recomendável utilizar um modelo de predição para cada tipo de estabelecimento, produto ou serviço.

Em trabalhos futuros, será investigado o impacto de diferentes técnicas de pré-processamento e conjuntos de atributos, como *part-of-speech* [Ott et al. 2011], métricas sobre a complexidade do texto dos *reviews* e sobre seus autores. Também, prevê-se a avaliação do MDLText em um cenário *online* e a realização de experimentos com um número maior de coleções para garantir mais evidência estatística nas análises *post-hoc*.

Agradecimentos

Os autores são gratos à agência de fomento CAPES (Proc. 1709642) pelo apoio financeiro a este projeto de pesquisa.

Referências

- Al Najada, H. e Zhu, X. (2014). iSRD: Spam review detection with imbalanced data distributions. In *Proceedings of the 15th International Conference on Information Reuse and Integration (IRI'14)*, pages 553–560, Redwood City, CA, USA. IEEE.
- Assis, F., Yerazunis, W., Siefkes, C., e Chhabra, S. (2006). Exponential differential document count – a feature selection factor for improving Bayesian filters accuracy. In *Proceedings of the 2006 MIT Spam Conference (SP'06)*, pages 1–6, Cambridge, MA, USA.
- Barron, A., Rissanen, J., e Yu, B. (1998). The minimum description length principle in coding and modeling. *IEEE Transaction on Information Theory*, 44(6):2743–2760.
- Boser, B. E., Guyon, I. M., e Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory (COLT'92)*, pages 144–152, Pittsburgh, PA, USA. ACM.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Breiman, L., Friedman, J., Stone, C. J., e Olshen, R. A. (1984). *Classification and regression trees*. CRC press.
- Cortes, C. e Vapnik, V. N. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.
- Cover, T. M. e Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transaction on Information Theory*, 13(1):21–27.
- Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., e Najada, H. A. (2015). Survey of review spam detection using machine learning techniques. *Journal of Big Data*, 42(7):3634–3642.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.
- Grünwald, P. D., Myung, I. J., e Pitt, M. A. (2005). *Advances in Minimum Description Length: Theory and Applications*. The MIT Press.

- Harris, C. G. (2012). Detecting deceptive opinion spam using human computation. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI'12*, page 87–93, Toronto, Canada.
- Hsu, C.-W. e Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.
- Jindal, N. e Liu, B. (2007). Review spam detection. In *Proceedings of the 16th international conference on World Wide Web (poster paper)*, WWW'07, page 1189–1190, Banff, Canada.
- Jindal, N. e Liu, B. (2008). Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM'08*, pages 219–229, Palo Alto, CA, USA.
- KC, S. e Mukherjee, A. (2016). On the temporal dynamics of opinion spamming: Case studies on yelp. In *Proceedings of the 25th International Conference on World Wide Web*, pages 369–379. International World Wide Web Conferences Steering Committee.
- Luca, M. e Zervas, G. (2015). Fake it till you make it: Reputation, competition, and yelp review fraud. *Management Science*, 62(12):3412–3427.
- Manning, C. D., Raghavan, P., e Schütze, H. (2009). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- McCallum, A. e Nigam, K. (1998). A comparison of event models for naive Bayes text classification. In *Proceedings of the 15th AAAI Workshop on Learning for Text Categorization (AAAI'98)*, pages 41–48, Madison, Wisconsin.
- Mukherjee, A., Venkataraman, V., Liu, B., e Glance, N. (2013). What yelp fake review filter might be doing? In *Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM'13*, pages 409–418, Cambridge, MA, USA.
- Ott, M., Cardie, C., e Hancock, J. T. (2013). Negative deceptive opinion spam. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT'11)*, pages 01–05, Atlanta, Georgia, USA. Association for Computational Linguistics.
- Ott, M., Choi, Y., Cardie, C., e Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT'11)*, pages 309–319, Portland, Oregon. Association for Computational Linguistics.
- Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465–471.
- Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431.
- Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The Smart retrieval system - experiments in automatic document processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ.
- Silva, R. M., Alberto, T. C., Almeida, T. A., e Yamakami, A. (2017a). Towards filtering undesired short text messages using an online learning approach with semantic indexing. *Expert Systems with Applications*, 83:314–325.
- Silva, R. M., Almeida, T. A., e Yamakami, A. (2017b). MDLText: An efficient and lightweight text classifier. *Knowledge-Based Systems*, 118:152–164.

Uma Abordagem de Combinação Não-Linear ARIMA-SVM para Previsão de Séries Temporais

Domingos S. de O. Santos Jr.¹, Leonardo Guimarães de Holanda¹,
Paulo S. G. de Mattos Neto^{1,2}

¹Escola Politécnica de Pernambuco, Universidade de Pernambuco (UPE)
{dsosj,lgh}@ecomp.poli.br

²Centro de Informática, Universidade Federal de Pernambuco (UFPE)
psgmn@cin.ufpe.br

Abstract. *Hybrid systems that combine Artificial Neural Networks with other forecasting methods have been widely used in the literature. Among these systems, architectures that use the error series, or residuals, have obtained relevant theoretical and practical results. In this context, this work proposes a hybrid system that employs an intelligent computing approach to non-linearly combine the time series forecasting with the residuals of this prediction. The hybrid system composes of ARIMA and SVM models reach relevant results in terms of accuracy, when compared with other hybrid systems of the literature.*

Resumo. *Sistemas híbridos que combinam Redes Neurais Artificiais com outros métodos de previsão têm sido largamente utilizadas na literatura. Dentre esses sistemas, arquiteturas que utilizam a série do erro de previsão, ou resíduos, têm alcançado resultados teóricos e práticos relevantes. Nesse contexto, esse trabalho propõe um sistema híbrido que utiliza uma abordagem de computação inteligente para combinar de forma não-linear a previsão da série temporal com os resíduos dessa previsão. O sistema híbrido composto pelos modelos ARIMA e SVM alcançam resultados relevantes em termos de acurácia, quando comparados com sistemas híbridos da literatura.*

1. Introdução

A análise e previsão de séries temporais é uma ferramenta importante em diversas áreas de conhecimento, como: Economia, Engenharia, Finanças, Geografia e Meteorologia [Box et al. 2008]. Nesse contexto, métodos de previsão de séries temporais são desenvolvidos para modelagem de uma variável temporal de interesse, visando alcançar uma alta performance em termos de acurácia [Adhikari and Agrawal 2013]. Dentre os modelos de análise e previsão de séries temporais, os métodos estatísticos clássicos, tais como: Autorregressivo (AR), Médias Móveis (MA) e Autorregressivo Integrado de Média Móvel (ARIMA) têm sido largamente utilizados na literatura [Box et al. 2008]. Esses modelos têm como pontos fortes suas características estatísticas e a metodologia de Box & Jenkins [Box et al. 2008], que é uma forma clara e coesa para determinar os parâmetros do modelo [Liu et al. 1992]. Entretanto, os métodos estatísticos clássicos têm como limitação a capacidade de modelar apenas os padrões lineares presentes na série temporal. Nesse sentido, séries temporais geradas por fenômenos naturais podem não ser bem modeladas por tais métodos, já que séries do mundo real geralmente apresentam padrões

não-lineares em sua estrutura. Desse modo, uma alternativa promissora para a análise e previsão de séries temporais são as abordagens baseadas em Redes Neurais Artificiais (RNAs), que têm como características: processamento não-linear, capacidade adaptativa e poder de generalização para dados faltantes e ruidosos [Khashei and Bijari 2010].

Contudo, de acordo com [Ginzburg and Horn 1993, Khashei and Bijari 2011, Hibon and Evgeniou 2005], existem problemas de previsão em que a utilização de apenas um modelo não é satisfatório, pelo fato que o domínio pode: (i) possuir alta complexidade, (ii) não ter informações suficientes, (iii) necessitar de acurácia elevada ou (iv) requerer que o modelo seja aplicado em vários problemas. Nesses casos a combinação de preditores, também chamado de sistemas híbridos (SH), podem ser uma abordagem interessante para mitigar tais limitações. Trabalhos da literatura indicam que a combinação de várias técnicas obtém maior robustez para as previsões, alcançando a capacidade de capturar padrões diferentes, culminando assim em uma menor variância dos resultados [Ghosh et al. 1992, Khashei and Bijari 2010].

Sistemas híbridos que empregam a combinação de RNAs com outros preditores para explorar padrões temporais contidos na série de erro¹ tem alcançado resultados promissores na literatura [Zhang 2003, Pai and Lin 2005, Khashei and Bijari 2010, Firmino et al. 2014, Firmino et al. 2015, de Mattos Neto et al. 2015, de Mattos Neto et al. 2017a, de Mattos Neto et al. 2017b]. Esse tipo de combinação é utilizada com o objetivo de modelar separadamente os padrões lineares e não-lineares da série temporal sob análise. Na literatura são encontradas duas suposições que guiam as abordagens de combinação do preditor da série e do preditor do erro: a da combinação linear [Pai and Lin 2005, Zhang 2003] e da combinação não-linear [Khashei and Bijari 2011, Aguilar et al. 2014]. Na primeira suposição, os trabalhos consideram que a combinação entre os preditores é feita a partir de uma soma [Zhang 2003, Pai and Lin 2005, Faruk 2010, Cadenas and Rivera 2010]. Na segunda suposição, a combinação entre o preditor da série e o preditor do erro é feita a partir de uma função não-linear [Chen and Wang 2007, Khashei and Bijari 2010, Khashei and Bijari 2011, Wongsathan and Seedadan 2016].

A partir da suposição da combinação linear, Zhang (2003) propôs um sistema híbrido que combina a modelagem linear do ARIMA com a modelagem não-linear da RNA do tipo *Multilayer Perceptron* (MLP), com o objetivo de capturar os diferentes padrões contidos na série temporal. A partir desse trabalho diversos sistemas híbridos foram propostos para diversas aplicações, como na previsão de: qualidade da água [Faruk 2010], preços de lenha produzidos de forma renovável [Koutroumanidis et al. 2009], velocidade do vento [Cadenas and Rivera 2010] e concentração de material particulado (*particulate matter* - PM_{10}) [de Mattos Neto et al. 2015]. Baseado na suposição de combinação linear, a fusão de outros métodos foram propostos, como por exemplo: ARIMA e Máquinas de Vetor de Suporte [Pai and Lin 2005] ou ARIMA e Rede Neural Autorregressiva Não-Linear (*Nonlinear Autoregressive Neural Network* - NARNN) [Zhou et al. 2014, Zhou et al. 2016].

Seguindo a suposição da combinação não-linear, alguns trabalhos [Khashei and Bijari 2010, Khashei and Bijari 2011] propõem um sistema híbrido

¹Também conhecida como resíduos, é obtida pela diferença entre o valor real da série temporal e a previsão de um dado modelo aplicado a esses dados.

que emprega uma RNA do tipo MLP com o objetivo de encontrar a melhor função de combinação. O sistema proposto pode ser dividido em duas etapas: primeiro um ARIMA é usado na modelagem da série temporal, depois disso a RNA do tipo MLP busca a melhor função que combina os valores da: previsão linear, série de erros e série temporal. A segunda etapa visa encontrar a função de combinação que obtém a melhor acurácia do sistema. Utilizando o pressuposto de [Khashei and Bijari 2011], quanto às relações entre os componentes lineares e não lineares das séries, [Aguilar et al. 2014] propõe a utilização de um modelo linear do tipo *Seasonal Autoregressive Integrated Moving Average* (SARIMA) combinado a uma rede MLP.

Dentre os trabalhos encontrados na literatura, duas propostas de combinação, uma linear e uma não-linear, alcançaram resultados promissores [Pai and Lin 2005, Khashei and Bijari 2011]. Na primeira, Pai e Lin (2005) combinaram linearmente os modelos ARIMA e SVM, alcançando resultados superiores em termos de acurácia na previsão de séries financeiras, quando comparados com outras abordagens da literatura. Na segunda, Khashei e Bijari (2011) combinaram não-linearmente os modelos ARIMA e MLP, alcançando resultados mais acurados que os modelos ARIMA, MLP e a combinação linear do ARIMA com MLP proposta por [Zhang 2003]. Nesse sentido, este artigo propõe o ARIMA*SVM, que combina os pontos fortes da abordagem não-linear proposta em [Khashei and Bijari 2011] com a acurácia da SVM quando utilizada na modelagem da série de erros [Pai and Lin 2005].

Neste trabalho resultados experimentais são apresentados para três séries temporais bastante utilizadas na literatura: Lince Canadense, Manchas Solares e Taxa de Câmbio UK/EUA. Os resultados são avaliados a partir de um conjunto de três métricas de desempenho e mostram que o método proposto alcança resultados competitivos, quando comparados com trabalhos da literatura.

A organização deste artigo é descrito a seguir: na Seção 2 são apresentadas as principais arquiteturas da literatura que utilizam o resíduo; na Seção 3, o modelo proposto é introduzido; na Seção 4 a metodologia dos experimentos, a base de dados e os resultados utilizando três métricas de desempenho são avaliados; e por fim, na Seção 5 são apresentadas as considerações finais desse trabalho.

2. Sistemas Híbridos que Utilizam a Série de Erro

As abordagens tradicionais de sistemas híbridos que empregam a combinação de preditores geralmente utilizam como fonte de informação a série do erro aditivo, que é calculada a partir da Equação 1,

$$E_t = Y_t - \hat{L}_t, \quad (1)$$

em que, Y_t é o valor real da série temporal e \hat{L}_t é a previsão gerada por um determinado modelo.

A partir da modelagem da série de erro, abordagens de combinação linear e não-linear podem ser utilizadas.

Uma diversa gama de modelos lineares e não-lineares podem ser empregados para a modelagem de Y_t , entretanto comumente os trabalhos da literatura [Zhang 2003, Pai and Lin 2005, Khashei and Bijari 2011, Faruk 2010,

Aguilar et al. 2014, Zhou et al. 2014] utilizam métodos lineares dada suas características estatísticas [Box et al. 2008, Zhang 2003].

2.1. Abordagem de Combinação Linear

A modelagem não-linear da série de erro resultante de uma modelagem linear foi inicialmente proposta por Zhang (2003). Nesse trabalho há a suposição que a série temporal é composta pela combinação linear de padrões lineares e não-lineares, descrita pela Equação 2.

$$Y_t = L_t + N_t, \quad (2)$$

em que L_t representa o componente linear e N_t o componente não-linear, que podem ser estimados a partir dos dados. Inicialmente, a previsão do modelo linear (\hat{L}) é gerada, sendo o erro E_t calculado a partir da Equação 1.

Dada as características estatísticas dos modelos lineares, como o ARIMA, a série de erro (E_t) não deve apresentar nenhum padrão linear. Então, é importante mencionar que, se o modelo linear foi bem especificado, nenhum teste estatístico linear, como Função de Autocorrelação (FAC), Ljung-Box, ou Box-Pierce, deverá indicar correlação linear na série de resíduos.

Como não existe mais padrões lineares relevantes em E_t , um método não-linear pode ser empregado com o objetivo de modelar possíveis padrões não-lineares ainda contidos no erro. Assim, a previsão dos padrões não-lineares (\hat{N}_t) pode ser realizada a partir da modelagem não-linear da série de erro.

A previsão final (\hat{Y}) do sistema híbrido é dada pela combinação linear (soma) da previsão do modelo linear (\hat{L}) com a previsão do modelo não-linear (\hat{N}) dada pela Equação 3.

$$\hat{Y}_t = \hat{L}_t + \hat{N}_t. \quad (3)$$

2.2. Abordagem de Combinação Não-Linear

As abordagens de combinação não-linear [Khashei and Bijari 2010, Khashei and Bijari 2011] supõem que a série temporal Y_t é composta por uma função não-linear entre os padrões linear e não-linear (Equação 4).

$$Y_t = f(L_t, N_t), \quad (4)$$

em que L_t e N_t são os componentes linear e não-linear, respectivamente, e $f(\cdot)$ é uma função que não é conhecida. Na primeira etapa, o modelo linear realiza a previsão (\hat{L}_t) e a série de erro é calculada a partir da Equação 1. Na segunda etapa da combinação, uma RNA do tipo MLP é utilizada com o objetivo de modelar a provável relação não-linear existente entre a série de erro (E_t), a previsão do modelo linear (\hat{L}_t) e os dados reais da série temporal (Y_t). A previsão do modelo não-linear é a previsão \hat{Y}_t descrita pela Equação 5,

$$\hat{Y}_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-n_1}, \hat{L}_t, e_{t-1}, e_{t-2}, \dots, e_{t-n_2}), \quad (5)$$

em que f representa a função do modelo não-linear que visa combinar a previsão do ARIMA (\hat{L}_t) com os valores passados de Y_t e E_t . Os valores inteiros n_1 e n_2 são determinados pela MLP e correspondem aos retardos temporais relevantes dos resíduos e_i ($i = 1, \dots, n_2$) e da série temporal y_j ($j = 1, \dots, n_1$), respectivamente.

O processo de desenvolvimento da modelagem não-linear (modelo MLP) tem a capacidade de remover qualquer variável de entrada (Y_t, \hat{L}_t, E_t) com o objetivo de encontrar o melhor relacionamento entre os componentes. Desse modo, a combinação busca encontrar o processo de geração de dados baseado nas estruturas lineares e não-lineares existentes na série temporal.

3. Método Proposto

O método proposto pode ser dividido em duas etapas: treinamento (a) e teste (b). A Figura 1 mostra a etapa (a), que consiste da estimação e/ou treinamento do método proposto. Essa etapa pode ser dividida em três módulos: (i) estimação do modelo linear, (ii) cálculo da série de erro, e (iii) treinamento do modelo não-linear.

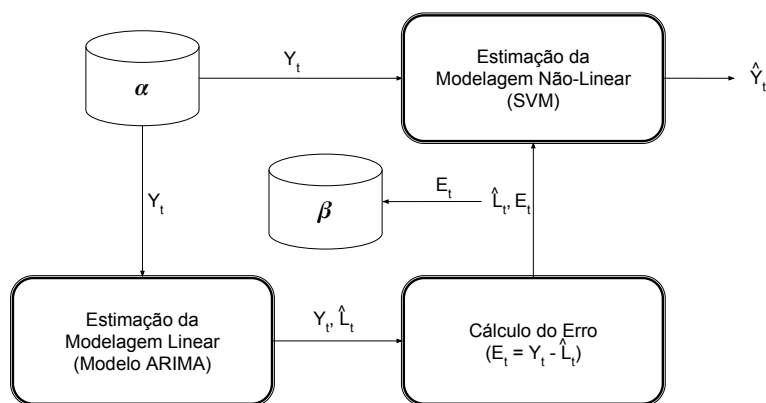


Figura 1. Etapa de treinamento do método proposto

O módulo (a.i) da Figura 1, Estimação da Modelagem Linear, recebe como entrada o conjunto de treinamento de uma dada série temporal (Y_t), armazenada na base de dados α , e gera como saída a previsão linear (\hat{L}_t). Nessa parte do treinamento, a estimação dos parâmetros da modelagem linear é realizada. Neste trabalho foi utilizado o modelo $ARIMA(p, d, q)$ e a metodologia de Box & Jenkins para a estimação dos seus parâmetros.

O módulo (a.ii) da etapa de treinamento calcula a série de erros para o conjunto de treinamento a partir da Equação 1. Esse módulo recebe como variáveis de entrada: os valores reais da série temporal (Y_t) e a previsão linear do modelo ARIMA (\hat{L}_t) do conjunto de treinamento. É importante mencionar que a base de dados β armazena a série de resíduos gerada no processo de treinamento.

O módulo (a.iii) realiza o treinamento da modelagem não-linear, combinando os retardos temporais da série temporal (Y_t), da série de resíduos (E_t) e a previsão do modelo ARIMA (\hat{L}_t). O objetivo do módulo (iii) é encontrar a melhor combinação, visando minimizar o erro entre a série temporal e a previsão do sistema híbrido.

Após o término do treinamento do sistema híbrido, os melhores parâmetros para a modelagem linear (ARIMA), para a modelagem não-linear (SVM), bem como a série de resíduos (E_t) são armazenados. A etapa de teste (b) do sistema híbrido, que pode ser vista na Figura 2, mostra o uso dos melhores parâmetros dos modelos encontrados na etapa de treinamento.

A etapa de teste (b) pode ser dividida em dois módulos: (i) previsão da modelagem linear (Modelo ARIMA), e (ii) previsão da modelagem não-linear (SVM). O módulo de previsão do modelo ARIMA (b.i) gera a previsão linear para o conjunto de teste (\hat{L}_t) a partir dos retardos temporais da série temporal (y_{t-1}, \dots, y_{t-n}). A quantidade de retardos temporais n depende do modelo ARIMA(p, d, q) encontrado na etapa de treinamento (a).

O módulo de previsão do modelo SVM (b.ii) recebe como entrada a previsão do modelo ARIMA (\hat{L}_t) e os retardos temporais da série temporal (y_{t-1}, \dots, y_{t-k}) e da série de erro (e_{t-1}, \dots, e_{t-m}). Os inteiros k e m representam o retardo temporal máximo usado para a série temporal e para a série de resíduos, respectivamente. A saída desse módulo é a previsão do valor futuro (\hat{Y}_t) para o conjunto de teste.

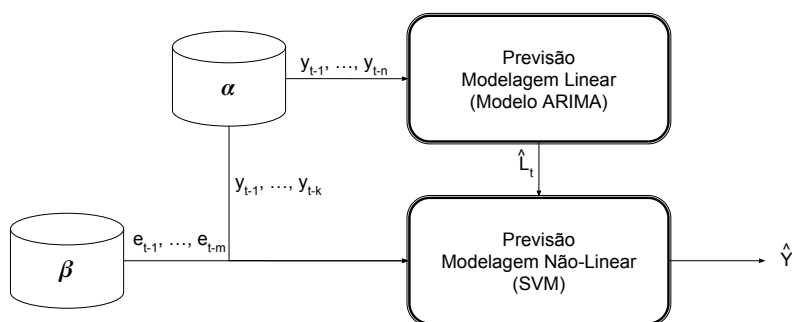


Figura 2. Etapa de teste do método proposto

4. Metodologia e Resultados Experimentais

A base de dados usada nos experimentos é composta por três séries temporais: Lince Canadense (*Canadian Lynx*), Manchas Solares (*Sunspot*) e Taxa de Câmbio da libra Britânica para dólar Americano (*British pound/US dollar exchange rate*). Essas séries temporais são provenientes de diferentes áreas de aplicação e apresentam características estatísticas distintas. Elas têm sido largamente estudadas na literatura de previsão de séries temporais [Zhang 2003, Khashei and Bijari 2010].

A série temporal de Lince Canadense contém a quantidade de lince presos por ano no distrito do rio Mackenzie, no norte do Canadá. Esse conjunto de dados é composto por 114 observações entre os anos de 1821 e 1934. A série de Manchas Solares consiste no registro da atividade anual de manchas visíveis no sol. Esse conjunto de dados contém o número de incidências anual de manchas solares entre 1700 e 1987, totalizando 288 observações. A série da Taxa de Câmbio da libra Britânica para dólar Americano contém as médias semanais entre 1980 e 1993, totalizando 731 registros. As séries de Lince e *Sunspot* estão disponíveis em [R Core Team 2016] e a série da Taxa de Câmbio está disponível em <https://fred.stlouisfed.org/series/DEXUSUK>.

A divisão dos conjuntos pode ser visualizada na Tabela 1 e segue os trabalhos [Zhang 2003, Khashei and Bijari 2011]. Baseado nas porcentagens descritas na Tabela 1, os conjuntos de treinamento e teste tem a seguinte quantidade de registros: 100 e 14 para Lince, 221 e 67 para *Sunspot* e 679 e 52 para Taxa de Câmbio. Na base Lince Canadense é empregada a suavização logarítmica na base 10. Na série de Taxa de Câmbio, utiliza-se a suavização logarítmica natural.

Tabela 1. Divisão dos conjuntos de treinamento e teste para a base de dados utilizada nos experimentos.

Série	Tamanho da Amostra	Treino (%)	Teste (%)
Lince	114	88	12
<i>Sunspot</i>	288	77	23
Taxa de Câmbio	731	93	7

As combinações lineares e não-lineares utilizadas nesse trabalho são representadas pelos símbolos “+” e “*”, respectivamente. Dessa forma, o método proposto, que é a combinação não-linear entre ARIMA e SVM é representado por ARIMA*SVM, enquanto a combinação linear é representada por ARIMA+SVM.

O método proposto tomou como base resultados já consolidados na literatura [Zhang 2003, Khashei and Bijari 2011]. Os parâmetros utilizados pelo modelo ARIMA(p,d,q) do método proposto são os mesmos utilizados pelos trabalhos da literatura [Zhang 2003, Khashei and Bijari 2011]. Para as séries Lince, *Sunspot* e Taxa de câmbio são utilizadas as seguintes configurações: ARIMA(12,0,0), ARIMA(9,0,0) e ARIMA(0,1,0), respectivamente.

Os parâmetros do SVM utilizado pelo método proposto (ARIMA*SVM) foram encontrados empiricamente a partir de 30 simulações para cada configuração. Para o SVM foi utilizado o *kernel* de base radial do pacote libsvm [Chang and Lin 2011]. As variáveis de custo, γ e número de entradas foram avaliados nos seguintes intervalos: 1 a 10000, $10e^{-6}$ a 1, e 1 a 20, respectivamente. Após a realização de experimentos variando os parâmetros nos intervalos supracitados, é selecionado para cada modelo, a configuração que alcança o menor MSE do treinamento. As configurações selecionadas são mostradas na Tabela 2.

Os modelos usados para comparação (MLP, ARIMA+MLP e ARIMA*MLP) utilizaram os mesmos parâmetros dos trabalhos [Zhang 2003, Khashei and Bijari 2011]. As configurações da RNA do tipo MLP foram as mesmas para os modelos MLP e ARIMA+MLP, seguindo a nomenclatura AxBxC, em que A é o número de neurônios na camada de entrada, B é o número de nodos na camada escondida e C é o número de neurônios na camada de saída. O algoritmo de treinamento utilizado foi o *Backpropagation*. As arquiteturas das MLP utilizadas para ARIMA+MLP e ARIMA*MLP para as séries Lince, *Sunspot* e Taxa de câmbio são, respectivamente: 7x5x1 e 8x3x1, 4x4x1 e 7x3x1, e 12x4x1. As entradas para o modelo MLP da combinação ARIMA*MLP [Khashei and Bijari 2011] são configuradas da seguinte maneira: $(e_{t-1}, e_{t-2}, \hat{L}_t, y_{t-1}, \dots, y_{t-5})$ para Lince, $(e_{t-1}, e_{t-2}, \hat{L}_t, y_{t-1}, \dots, y_{t-4})$ para *Sunspot* e $(e_{t-1}, \dots, e_{t-6}, \hat{L}_t, y_{t-1}, \dots, y_{t-5})$ para Taxa de Câmbio. Todos os experimentos foram desenvolvidos em software R [R Core Team 2016].

4.1. Medidas de Desempenho

Os resultados dos experimentos são avaliados a partir de três medidas de desempenho bastante utilizadas na literatura [Zhang 2003, Khashei and Bijari 2010]: erro médio quadrático (*Mean Square Error* - MSE) descrito na Equação 6, erro percentual médio absoluto (*Mean Absolute Percentage Error* - MAPE) descrito na Equação 7 e desvio médio absoluto (*Mean Absolute Deviation* - MAD) descrito na Equação 8.

Tabela 2. Configurações das SVMs utilizadas para cada série temporal.

Lince		
Config.	ARIMA*SVM	ARIMA+SVM
Entradas	$y_{t-6}, \dots, y_{t-1}, e_{t-6}, \dots, e_{t-1}, L_t$	e_{t-2}, e_{t-1}
Custo	1000	1500
Gamma	0.0001	0.1
Sunspot		
Config.	ARIMA*SVM	ARIMA+SVM
Entradas	$y_{t-8}, \dots, y_{t-1}, e_{t-6}, \dots, e_{t-1}, L_t$	e_{t-6}, \dots, e_{t-1}
Custo	10000	1
Gamma	0.0001	0.1
Taxa de Câmbio		
Config.	ARIMA*SVM	ARIMA+SVM
Entradas	$y_{t-16}, \dots, y_{t-1}, e_{t-6}, \dots, e_{t-1}, L_t$	e_{t-8}, \dots, e_{t-1}
Custo	0.1	50
Gamma	0.01	1

Para as Equações 6, 7 e 8, a variável N representa o tamanho do conjunto de dados, y_t representa o valor real da série no tempo t e \hat{y}_t é a previsão para y_t . Para todas as métricas selecionadas, quanto menor o seu valor, melhor é a acurácia do modelo.

$$MSE = \frac{1}{N} \sum_{t=1}^N (y_t - \hat{y}_t)^2, \quad (6)$$

$$MAPE = \frac{100}{N} \sum_{t=1}^N \left| \frac{y_t - \hat{y}_t}{y_t} \right|, \quad (7)$$

$$MAD = \frac{1}{N} \sum_{t=1}^N |y_t - \hat{y}_t|, \quad (8)$$

4.2. Resultados

Nas Tabelas 3, 4 e 5, são exibidos os melhores resultados encontrados para as séries de Lince, *Sunspot* e Taxa de Câmbio, utilizando três medidas de desempenho (MSE, MAPE e MAD). Para cada série temporal estudada, o melhor resultado de cada medida de desempenho está destacado em negrito.

Na Tabela 3 podem ser vistos os resultados para a série de Lince. É possível observar que os modelos que utilizam combinação (linear e não-linear) alcançaram melhores resultados em termos de MSE e MAD, que os modelos simples, ARIMA e MLP. Dentre os modelos de combinação linear investigados, o modelo ARIMA+SVM [Pai and Lin 2005] alcançou o melhor resultado para todas as métricas investigadas. É importante destacar que em termos de MAD, o modelo ARIMA+SVM [Pai and Lin 2005] obteve 0,0888, que foi uma ordem de grandeza menor que o erro alcançado pelo ARIMA+MLP [Zhang 2003] que foi de 0,1040. Entre os modelos combinados, na Tabela 3 pode ser visto que a abordagem de combinação não-linear (ARIMA*MLP e ARIMA*SVM) alcançou os melhores resultados em, pelo menos, duas métricas de desempenho das três utilizadas. A combinação não-linear entre ARIMA e

Tabela 4. Performance dos melhores modelos encontrados para a série *Sunspot*.

Abordagem	Método	Referência	MSE	MAPE	MAD
Modelo Simples	ARIMA	[Zhang 2003]	306,08	–	13,033
	MLP	[Zhang 2003]	351,19	–	13,544
Combinação Linear	ARIMA+MLP	[Zhang 2003]	280,15	33,70	12,780
	ARIMA+SVM	[Pai and Lin 2005]	288,09	31,19	12,630
Combinação Não-Linear	ARIMA*MLP	[Khashei and Bijari 2011]	218,64	27,21	11,446
	ARIMA*SVM	Método Proposto	314,36	27,67	12,346

SVM (ARIMA*SVM) obteve os melhores resultados em termos de MSE e MAPE, enquanto que o modelo ARIMA*MLP proposto por [Khashei and Bijari 2011] alcançou o melhor desempenho em termos do MAD.

Tabela 3. Performance dos melhores modelos encontrados para a série *Lince Canadense*.

Abordagem	Método	Referência	MSE	MAPE	MAD
Modelos Simples	ARIMA	[Zhang 2003]	0,0204	–	0,1123
	MLP	[Zhang 2003]	0,0204	–	0,1121
Combinação Linear	ARIMA+MLP	[Zhang 2003]	0,0172	3,36	0,1040
	ARIMA+SVM	[Pai and Lin 2005]	0,0126	2,95	0,0888
Combinação Não-Linear	ARIMA*MLP	[Khashei and Bijari 2011]	0,0099	2,83	0,0850
	ARIMA*SVM	Método Proposto	0,0077	2,23	0,0965

Na Tabela 4 é possível observar os resultados dos modelos simples e das abordagens de combinação para a série *Sunspot*. Pode ser observado que os modelos de combinação linear alcançaram melhores resultados que os modelos simples em termos de MSE e MAD. Na comparação entre as abordagens de combinação, o modelo ARIMA*MLP alcançou o melhor resultado para todas as medidas de desempenho investigadas. O modelo proposto por Khashei e Bijari (2011) alcançou resultados bem superiores que os demais modelos em termos de MSE. O método proposto, ARIMA*SVM, apesar de ter obtido um desempenho inferior que o modelo ARIMA e as demais abordagens de combinação em termos de MSE, alcançou melhores resultados que a combinação linear e os modelos simples em termos de MAPE e MAD. Esse fenômeno pode ocorrer quando diferentes métricas são utilizadas, como reportado em [Silva et al. 2014, de Mattos Neto et al. 2015, Junior et al. 2010]. Dada a diferença de abordagens entre as medidas de desempenho, ao se ter uma melhor performance em uma métrica não necessariamente essa melhoria é transferida para as demais métricas utilizadas. Mesmo assim, o ARIMA*SVM alcançou o segundo melhor resultado em termos de MAPE e MAD. Então, é possível afirmar que a abordagem não-linear é mais acurada para a série *Sunspot*.

Na Tabela 5 os resultados para o conjunto de teste da Taxa de Câmbio podem ser avaliados. Em termos de MSE, as abordagens de combinação alcançaram resultados superiores em relação ao modelo ARIMA. Em relação ao modelo simples MLP, as combinações que utilizaram a MLP para a modelagem do resíduo obtiveram resultados

Tabela 5. Performance dos melhores modelos encontrados para a série Taxa de Câmbio.

Abordagem	Método	Referência	MSE	MAPE	MAD
Modelo Simples	ARIMA	[Zhang 2003]	2,9392	–	0,0139
	MLP	[Zhang 2003]	2,8495	–	0,0135
Combinação Linear	ARIMA+MLP	[Zhang 2003]	2,9353	3,45	0,0139
	ARIMA+SVM	[Pai and Lin 2005]	2,7332	3,20	0,0129
Combinação Não-Linear	ARIMA*MLP	[Khashei and Bijari 2011]	2,8953	3,44	0,0139
	ARIMA*SVM	Método Proposto	2,7084	3,47	0,0139

Tabela 6. Diferença percentual entre os modelos simples e de combinação com relação ao MSE. Os melhores resultados estão destacados em negrito.

Série Temporal		Lince		<i>Sunspot</i>		Taxa de Câmbio	
Abordagem	Método	ARIMA	MLP	ARIMA	MLP	ARIMA	MLP
Combinação Linear	ARIMA+MLP	16	16	8	20	0	-3
	ARIMA+SVM	38	38	6	18	7	4
Combinação Não-Linear	ARIMA*MLP	51	51	29	38	1	-2
	ARIMA*SVM	66	66	-3	10	8	5

levemente piores. O ARIMA*SVM obteve o melhor resultado considerando a métrica MSE. Em relação ao MAPE e MAD, é possível visualizar na Tabela 5 que o modelo ARIMA+SVM alcança o melhor resultado, enquanto que as outras combinações têm desempenhos similares. Se apenas o MAD for considerado, os dois melhores modelos foram o ARIMA+SVM e MLP, respectivamente.

A Tabela 6 mostra o ganho percentual dos métodos de combinação (linear e não-linear) em relação aos modelos simples (ARIMA e RNA do tipo MLP) sob a luz do MSE. É possível verificar que para os três conjuntos de dados os maiores ganhos foram obtidos pela combinação não-linear. O modelo proposto nesse trabalho (ARIMA*SVM) alcançou os melhores resultados em termos de MSE para as séries Lince e Taxa de Câmbio, enquanto que o modelo proposto em [Khashei and Bijari 2011], ARIMA*MLP, alcançou o melhor resultado para a série de *Sunspot*. Essa diferença pode estar relacionada com a limitação do modelo, com o processo de especificação dos seus parâmetros, ou a natureza da série temporal.

5. Considerações Finais

Este artigo apresenta uma nova abordagem de sistema híbrido que combina de forma não-linear o modelo ARIMA (formalismo linear) com uma SVM (formalismo não-linear). A combinação é feita de tal modo que o SVM possa modelar e prever padrões não-lineares contidos na série erro. A abordagem desenvolvida utiliza como base dois trabalhos da literatura que têm recebido grande destaque dado a acurácia dos seus modelos [Khashei and Bijari 2011, Pai and Lin 2005].

O modelo proposto é avaliado e comparado com resultados do estado da arte da literatura a partir de três métricas de desempenho: MSE, MAPE e MAD. A base de dados utilizada engloba três séries temporais com características distintas: Lince Canadense,

Manchas Solares (*Sunspot*) e Taxa de Câmbio UK/EUA. Para a série de Lince Canadense, o modelo proposto obteve os melhores resultados em termos de MSE e MAPE, quando comparado com os modelos da literatura. Para a série de *Sunspot*, o método proposto alcançou o segundo melhor desempenho em termos de MAPE e MAD, sendo inferior nessas métricas apenas para a modelagem não linear proposta em [Khashei and Bijari 2011]. Para a série de Taxa de Câmbio, o ARIMA*SVM alcançou o melhor desempenho em termos de MSE, sendo a melhor modelagem não-linear para essa base de dados. Em termos percentuais o método proposto alcança, em termos de MSE, a melhoria mais significativa para as séries de Lince e Taxa de Câmbio. Para trabalhos futuros é importante investigar a viabilidade de uma etapa exclusiva para tratamento, análise e previsão da série de resíduos. Nesse sentido, também é interessante estudar diferentes estratégias para combinar as previsões da série temporal com a informação que pode ser capturada da série de resíduos. Além disso, outros tipos de RNAs podem ser investigadas, tais como *Extreme Learning Machine* e *Deep Learning*.

Referências

- Adhikari, R. and Agrawal, R. (2013). An Introductory Study on Time Series Modeling and Forecasting. *arXiv*, 1302.6613.
- Aguilar, R. J., Turias, I., and Come, J. M. (2014). Hybrid approaches based on sarima and artificial neural networks for inspection time series forecasting. *Transportation Research Part E: Logistics and Transportation Review*, 67:1–13.
- Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2008). *Time series analysis: forecasting and control*. John Wiley & Sons.
- Cadenas, E. and Rivera, W. (2010). Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA–ANN model. *Renewable Energy*, 35(12):2732–2738.
- Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.
- Chen, K.-Y. and Wang, C.-H. (2007). A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan. *Expert Systems with Applications*, 32(1):254–264.
- de Mattos Neto, P., Cavalcanti, G., Madeiro, F., and Ferreira, T. (2015). An approach to improve the performance of PM forecasters. *PLoS ONE*, 10(9):e0138507.
- de Mattos Neto, P. S., Cavalcanti, G. D., and Madeiro, F. (2017a). Nonlinear combination method of forecasters applied to pm time series. *Pattern Recognition Letters*.
- de Mattos Neto, P. S., Ferreira, T. A., Lima, A. R., Vasconcelos, G. C., and Cavalcanti, G. D. (2017b). A perturbative approach for enhancing the performance of time series forecasting. *Neural Networks*, 88:114 – 124.
- Faruk, D. Ö. (2010). A hybrid neural network and ARIMA model for water quality time series prediction. *Engineering Applications of Artificial Intelligence*, 23(4):586–594.
- Firmino, P. R. A., de Mattos Neto, P. S., and Ferreira, T. A. (2014). Correcting and combining time series forecasters. *Neural Networks*, 50(0):1 – 11.
- Firmino, P. R. A., de Mattos Neto, P. S., and Ferreira, T. A. (2015). Error modeling approach to improve time series forecasters. *Neurocomputing*, 153:242–254.

- Ghosh, J., Beck, S. D., and Chu, C.-C. (1992). Evidence combination techniques for robust classification of short-duration oceanic signals. In *Aerospace Sensing*, pages 266–276. International Society for Optics and Photonics.
- Ginzburg, I. and Horn, D. (1993). Combined Neural Networks for Time Series Analysis. In *7th NIPS Conference - Advances in Neural Information Processing Systems 6*, pages 224–231.
- Hibon, M. and Evgeniou, T. (2005). To combine or not to combine: selecting among forecasts and their combinations. *International Journal of Forecasting*, 21(1):15–24.
- Junior, A. R. L., Silva, D. A., de Mattos Neto, P. S. G., and Ferreira, T. A. E. (2010). An experimental study of fitness function and time series forecasting using artificial neural networks. In *Genetic and Evolutionary Computation Conference, GECCO 2010, Proceedings, Portland, Oregon, USA, July 7-11, 2010, Companion Material*, pages 2015–2018.
- Khashei, M. and Bijari, M. (2010). An artificial neural network (p, d, q) model for time-series forecasting. *Expert Systems with applications*, 37(1):479–489.
- Khashei, M. and Bijari, M. (2011). A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Applied Soft Computing*, 11(2):2664–2675.
- Koutroumanidis, T., Ioannou, K., and Arabatzis, G. (2009). Predicting fuelwood prices in Greece with the use of ARIMA models, artificial neural networks and a hybrid ARIMA–ANN model. *Energy Policy*, 37(9):3627–3634.
- Liu, L.-M., Hudak, G. B., Box, G. E., Muller, M. E., and Tiao, G. C. (1992). *Forecasting and time series analysis using the SCA statistical system*, volume 1. Scientific Computing Associates DeKalb, IL.
- Pai, P.-F. and Lin, C.-S. (2005). A hybrid ARIMA and Support Vector Machines model in stock price forecasting. *Omega*, 33(6):497–505.
- R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Silva, D. A., Alves, G. I., de Mattos Neto, P. S., and Ferreira, T. A. (2014). Measurement of fitness function efficiency using data envelopment analysis. *Expert Systems with Applications*, 41(16):7147 – 7160.
- Wongsathan, R. and Seedadan, I. (2016). A Hybrid ARIMA and Neural Networks Model for PM-10 Pollution Estimation: The Case of Chiang Mai City Moat Area. *Procedia Computer Science*, 86:273–276.
- Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50:159–175.
- Zhou, L., Xia, J., Yu, L., Wang, Y., Shi, Y., Cai, S., and Nie, S. (2016). Using a Hybrid Model to Forecast the Prevalence of Schistosomiasis in Humans. *International Journal of Environmental Research and Public Health*, 13(4):355.
- Zhou, L., Yu, L., Wang, Y., Lu, Z., Tian, L., Tan, L., Shi, Y., Nie, S., and Liu, L. (2014). A hybrid model for predicting the prevalence of schistosomiasis in humans of qianjiang city, china. *PLoS ONE*, 9(8):e104875.

Uma Nova Variante do Algoritmo do Morcego Baseada em uma Modificação no Operador de Mutação de Michalewicz

Izabele V. O. Leite¹, Marcos H. F. Marcone², Fábio A. P. Paiva³

¹ Departamento de Engenharia Química
Universidade Federal do Rio Grande do Norte (UFRN) – Natal – RN – Brasil

² Escola de Ciências e Tecnologia
Universidade Federal do Rio Grande do Norte (UFRN) – Natal – RN – Brasil

³ Diretoria Acadêmica
Instituto Federal do Rio Grande do Norte (IFRN) – Parnamirim – RN – Brasil

{izaleite11, marcosmarcone48}@hotmail.com, fabio.procopio@ifrn.edu.br

Abstract. *Metaheuristics have been widely used to solve optimization problems. In this paper, a new variant of the metaheuristic inspired by echolocation of bats is presented. The proposed variant is the result of the combination between the original Bat Algorithm (BA) and a modification in the Michalewicz mutation operator. To evaluate the performance of this new variant, it was compared to the original BA and a variant of the literature named Modified BA (MBA). The computational experiments were performed with 30 independent runs to optimize six benchmark functions. The superiority of the proposed variant has been evidenced when its results are compared to the original BA and to the MBA variant results.*

Resumo. *As meta-heurísticas vêm sendo bastante utilizadas para resolver problemas de otimização. Neste trabalho, é apresentada uma nova variante da meta-heurística inspirada na ecolocalização dos morcegos. A variante proposta é o resultado da combinação entre o Bat Algorithm (BA) original e uma modificação no operador de mutação de Michalewicz. Para avaliar o desempenho da nova variante, ela foi comparada com o BA original e com uma variante da literatura chamada Modified BA (MBA). Os experimentos computacionais foram realizados com 30 execuções independentes a fim de otimizar seis funções de referência. A superioridade da variante proposta foi evidenciada ao comparar os seus resultados com os do BA original e com os do MBA.*

1. Introdução

Os problemas de otimização estão presentes em diversas aplicações reais como telecomunicações [Silva and Martins 2013], sistemas de potência [Chiaradonna et al. 2015], sinalização de tráfego [Matos and Carvalho 2016] e muitos outros. A otimização é um processo que visa encontrar a melhor solução para um determinado problema. Assim, há uma necessidade constante em se desenvolver algoritmos eficientes e robustos capazes de resolver problemas reais. Para resolver problemas desse tipo, os algoritmos meta-heurísticos têm sido amplamente utilizados.

As meta-heurísticas podem ser usadas para orientar uma heurística a fim de diversificar e de intensificar o espaço de busca do problema. Algumas meta-heurísticas se

inspiraram na natureza e, como um exemplo, podem ser citadas aquelas que se baseiam na inteligência de enxames. Por conta disso, elas são conhecidas como algoritmos baseados na inteligência de enxames. Normalmente, os sistemas de inteligência de enxame são implementados por uma população de agentes simples que interagem uns com os outros e com o seu próprio ambiente. A inteligência de enxames implementa a coletividade de grupos formados por agentes da natureza como pássaros, vaga-lumes, morcegos e outros.

Nos últimos anos, várias meta-heurísticas baseadas na inteligência de enxames têm sido propostas. O Algoritmo do Morcego (*Bat Algorithm* – BA) [Yang 2010], por exemplo, é inspirado na ecolocalização dos morcegos e, para equilibrar a diversificação e a intensificação, ele usa uma técnica de ajuste de frequência. O BA foi usado para resolver diversos problemas reais como veículos aéreos não-tripulados [Wang et al. 2012], classificação de dados [Mishra et al. 2012] e sistemas de distribuição [Reddy and Manoj 2012].

Um problema comumente enfrentado pelos algoritmos meta-heurísticos é conhecido como convergência prematura e ela ocorre quando o algoritmo fica “preso” em um ótimo local. A meta-heurística baseada na ecolocalização dos morcegos é poderosa em buscas locais, porém, às vezes, ela pode “cair” em um ótimo local e não conseguir “escapar” fazendo com que não realize uma boa busca global.

Vários trabalhos foram apresentados a fim de melhorar o desempenho do BA original. O *Hybrid Bat Algorithm* (HBA) [Fister Jr et al. 2013] é o resultado de uma hibridação do BA com as estratégias da Evolução Diferencial [Storn and Price 1997]. Em [Yilmaz and Kucuksille 2013], é apresentado o *Improved Bat Algorithm* com três modificações: Modificação no Fator de Peso de Inércia, Modificação da Frequência Adaptativa e Modificação da Abelha Escoteira. A fim de melhorar o mecanismo de exploração do BA original, o *Modified Bat Algorithm* (MBA) [Yilmaz et al. 2014] propõe uma modificação nas equações que modelam a taxa de emissão de pulso e a sonoridade de cada morcego. Em [Wang et al. 2012] e em [Zhang and Wang 2012], as propostas usaram estratégias de mutação da Evolução Diferencial [Storn and Price 1997] como uma alternativa para melhorar o algoritmo original.

Este trabalho utiliza a mutação de Michalewicz [Michalewicz 1996] a fim de apresentar uma variante da meta-heurística inspirada na ecolocalização dos morcegos. O operador de mutação garante que, no início do processo de otimização, o algoritmo realize buscas globais e, ao longo da evolução das iterações, as buscas passem a ser locais. Para melhorar a performance do BA original, o operador de Michalewicz foi modificado a fim de que a mutação ocorra em todas as dimensões do vetor que representa a posição do morcego, ao invés de apenas uma dimensão. Seis funções de referência são usadas para comparar a variante proposta e o BA original. Após os experimentos, os resultados mostraram que o desempenho da variante apresentada superou o do BA.

O trabalho está organizado da seguinte forma: na Seção 2, é apresentada uma breve introdução sobre a meta-heurística BA e o operador de mutação de Michalewicz; na Seção 3, a nova variante é apresentada para melhorar a performance do BA original. Na Seção 4, os experimentos são apresentados e os resultados são discutidos. Por fim, a Seção 5 apresenta as considerações finais e os trabalhos futuros.

2. Fundamentação Teórica

Aqui são apresentadas algumas informações necessárias para o entendimento da variante proposta na Seção 3, como a meta-heurística inspirada na ecolocalização de morcegos e o operador de mutação de Michalewicz.

2.1. Metaheurística Inspirada na Ecolocalização de Morcegos

Os algoritmos meta-heurísticos bioinspirados têm recebido muita atenção por parte dos cientistas da computação. Em [Yang 2010], é proposto um desses algoritmos o qual é inspirado na ecolocalização de morcegos e, por isso, é conhecido como *Bat Algorithm* (BA). A ecolocalização é uma sofisticada capacidade biológica que os morcegos utilizam para detectar presas e evitar obstáculos. Durante o voo, os morcegos emitem ondas sonoras e ouvem ecos para, assim, construir um mapa sônico para representar suas proximidades. Dessa forma, eles podem calcular o quão longe estão de um determinado objeto e, além disso, eles são capazes de distinguir um obstáculo de uma presa, mesmo no escuro [Nakamura et al. 2012].

Na meta-heurística BA, todos os morcegos usam a ecolocalização para calcular distâncias. O i -ésimo morcego possui uma posição x_i (vetor solução), uma velocidade v_i e uma frequência f_i . Para cada iteração t , o i -ésimo morcego se move em direção do melhor morcego encontrado até o momento. A fim de implementar a busca global, são atualizadas a posição, a velocidade e a frequência do i -ésimo morcego, de acordo com a Equação (1):

$$\begin{aligned} f_i &= f_{min} + (f_{max} - f_{min}) \cdot \beta \\ v_i^t &= v_i^{t-1} + (x_i^{t-1} - x_g^{t-1}) \cdot f_i \\ x_i^t &= x_i^{t-1} + v_i^t, \end{aligned} \quad (1)$$

onde f_{min} e f_{max} são, respectivamente, as frequências mínima e máxima. β é um valor aleatório gerado a partir de uma distribuição uniforme $\in [0, 1]$ e x_g^{t-1} representa a melhor solução global corrente.

Já a busca local é implementada por meio da geração de uma nova solução candidata próxima da melhor solução atual, como segue:

$$x_{nova}^t = x_g^{t-1} + \epsilon \bar{A}^t, \quad (2)$$

onde ϵ é um valor aleatório $\in [-1, 1]$ e \bar{A}^t é o valor médio do som produzido por todos os morcegos. Além da velocidade e da frequência, as ondas sonoras emitidas pelos morcegos possuem outras duas características: a taxa de pulso e a sonoridade. A taxa de pulso é a medida da variação das perturbações que se propagam através de um meio. Já a sonoridade, também conhecida como amplitude, é a magnitude de oscilação de uma onda [Cordeiro et al. 2012]. Quando um morcego identifica uma presa, de acordo com a sua proximidade, a sua taxa de pulso (r) é aumentada e sua sonoridade (A) é diminuída. No BA, implementa-se a sonoridade $A_i \in [A_0, A_{min}]$ e a taxa $r_i \in [0, 1]$, como segue:

$$\begin{aligned} A_i^{t+1} &= \alpha A_i^t \\ r_i^{t+1} &= r_i^0 [1 - \exp(-\gamma t)], \end{aligned} \quad (3)$$

onde α e γ são valores constantes. Para qualquer $0 < \alpha < 1$ e $\gamma > 0$, tem-se

$$\begin{aligned} A_i^t &\rightarrow 0 \\ r_i^t &\rightarrow r_i^0, \text{ onde } t \rightarrow \infty. \end{aligned} \tag{4}$$

O pseudocódigo do BA pode ser sumarizado pelo Algoritmo 1. Ele é inicializado com uma geração aleatória da população de morcegos (linha 1). Na linha 2, a frequência inicial f_i é determinada para a posição X_i . A linha 3 inicializa a taxa de pulso r_i e a sonoridade A_i . As linhas 4–16 representam a evolução dos morcegos ao longo do tempo. Nesta evolução, novas soluções são estabelecidas e a frequência, a velocidade e a posição de cada morcego são atualizadas (linha 6). Na próxima linha, a taxa de emissão de pulsos é comparada com um número real aleatório gerado a partir de uma distribuição uniforme no intervalo $[0, 1]$. Quando o número gerado for maior que r_i , novas soluções são geradas em torno da melhor solução (linha 9). Na linha 11, a solução atual é avaliada e é verificado se ela será aceita. Em caso positivo, a taxa r_i é aumentada e a sonoridade A_i é diminuída (linha 12). Por fim, o bando é ordenado e o melhor morcego é selecionado (linha 14).

Algoritmo 1 Pseudocódigo do Algoritmo do Morcego

- 1: Inicializa a população de morcegos X_i $i = (1, 2, \dots, N)$ e V_i
 - 2: Define a frequência de pulso f_i em X_i
 - 3: Inicialize a taxa de pulso r_i e a sonoridade A_i
 - 4: **while** ($t \leq num_max_iter$) **do**
 - 5: **for** $i = 1 : X$ **do**
 - 6: Gere novas soluções ajustando frequência e atualizando velocidades e posições
 - 7: **if** ($rand > r_i$) **then**
 - 8: Selecione a melhor solução
 - 9: Gere uma solução nas proximidades da melhor solução
 - 10: **end if**
 - 11: **if** ($rand < A_i \ \& \ fit(x_i) < fit(x_g^{t-1})$) **then**
 - 12: Aumente r_i e diminua A_i
 - 13: **end if**
 - 14: Classifique os morcegos e encontre o melhor
 - 15: **end for**
 - 16: **end while**
-

2.2. Operador de Mutação de Michalewicz

O processo de mutação consiste na geração de novas soluções a partir de uma solução inicial. Assim, dada uma solução $x \in \mathfrak{R}$, o valor correspondente de x mutado pode ser expresso por $x' = mut(x)$, onde $mut(\cdot)$ é a função de mutação.

Os operadores de mutação são parte integrante das técnicas de computação evolutiva e podem ser usados para evitar a perda da diversidade de uma população. Eles podem introduzir novos indivíduos a uma população adicionando variabilidade à população e impedindo a estagnação em ótimos locais [Dumitrescu et al. 2000]. Assim, a inclusão de um operador de mutação ao BA pode aumentar a sua capacidade de busca.

Michalewicz desenvolveu um operador de mutação [Michalewicz 1996] para ser usado, inicialmente, em um Algoritmo Genético, o qual é chamado de operador de mutação não-uniforme de Michalewicz. O operador pode ser usado para trabalhar modificando uma dimensão do vetor que representa a posição do morcego x_{id} , como mostrado

$$mut(x_{id}) = \begin{cases} x_{id} + \Delta(t, u - x_{id}) : & b_a = 1 \\ x_{id} + \Delta(t, x_{id} - l) : & b_a = 0, \end{cases} \quad (5)$$

onde t é o número da iteração atual, u e l são os limites superior e inferior do valor da dimensão do morcego, respectivamente; b_a é um bit gerado aleatoriamente. A função $\Delta(t, y)$ retorna um valor no intervalo $[0, y]$, tal que a probabilidade de $\Delta(t, y)$ ser próximo de 0 aumenta à medida que t aumenta. Essa propriedade faz com que, nas primeiras iterações (isto é, quando t é pequeno), o algoritmo realize buscas globais, ao passo que quando t aumenta, as buscas passam a ser locais. A função $\Delta(t, y)$ é dada por:

$$\Delta(t, y) = y \left(1 - a^{(1-t/T)^b} \right), \quad (6)$$

onde a é um número aleatório gerado a partir de uma distribuição uniforme no intervalo $[0, 1]$, T é o número máximo de iterações do algoritmo e b é um parâmetro de tunelamento, cujo valor padrão é 5, como descrito em [Michalewicz 1996].

3. Uma Nova Variante BA Baseada no Operador Modificado de Michalewicz

O Algoritmo do Morcego é uma técnica capaz de encontrar bons resultados em um tempo de convergência rápido, porém ele pode convergir prematuramente. Diz-se que o algoritmo convergiu prematuramente quando a solução que ele propõe se aproxima de um ótimo local, ao invés do ótimo global. A convergência prematura ocorre devido à diminuição da diversidade no espaço de busca [Evers and Ghalia 2009], o que leva o algoritmo a um estado de estagnação.

Para reduzir os problemas que podem ser causados pela convergência prematura, este trabalho propõe a combinação do operador de mutação de Michalewicz com o Algoritmo do Morcego. No entanto, ao contrário da abordagem original do operador, que altera apenas uma dimensão do vetor (x_{id}), como visto na Equação 5, neste trabalho, todas as dimensões do vetor x_i são alteradas, como segue:

$$mut(x_i) = \begin{cases} x_i + \Delta(t, U - x_i) : & num_{sorteado} = 0 \\ x_i + \Delta(t, x_i - L) : & num_{sorteado} = 1, \end{cases} \quad (7)$$

onde t é o número da iteração atual, U e L são os vetores que correspondem aos limites superior e inferior de cada dimensão do morcego, respectivamente; $num_{sorteado}$ tem 50% de probabilidade de ser 0 e 50% de probabilidade de ser 1; $\Delta(t, y)$ é uma função que retorna um valor no intervalo $[0, y]$ e pode ser calculada a partir da Equação 6.

O pseudocódigo do BA baseado no operador modificado de Michalewicz pode ser resumido pelo Algoritmo 2. A contribuição do trabalho é apresentada nas linhas 11–13. As demais são semelhantes ao Algoritmo 1 e já foram descritas na Subseção 2.1. Na linha 11, o valor de *fitness* do morcego atual é comparado com o seu valor de *fitness* quando a mutação de Michalewicz é aplicada. Então, caso essa mutação apresente um valor de

fitness melhor que o valor de *fitness* original, a mutação é confirmada e o morcego atual assume a sua nova posição (linha 12). A verificação na linha 11 é realizada para que a aplicação da mutação só seja efetivada se houver uma melhoria no valor de *fitness*.

Algoritmo 2 Pseudocódigo do Algoritmo do Morcego Modificado com Michalewicz

```

1: Inicializa a população de morcegos  $X_i$   $i = (1, 2, \dots, N)$  e  $V_i$ 
2: Define a frequência de pulso  $f_i$  em  $X_i$ 
3: Inicialize a taxa de pulso  $r_i$  e a sonoridade  $A_i$ 
4: while ( $t \leq num\_max\_iter$ ) do
5:   for  $i = 1 : X$  do
6:     Gere novas soluções ajustando frequência e atualizando velocidades e posições
7:     if ( $rand > r_i$ ) then
8:       Selecione a melhor solução
9:       Gere uma solução nas proximidades da melhor solução
10:    end if
11:    if  $fit(mutacao\_michalewicz(x_i)) < fit(x_i)$  then
12:       $x_i \leftarrow mutacao\_michalewicz(x_i)$ , conforme Equação 7
13:    end if
14:    if ( $rand < A_i \ \& \ fit(x_i) < fit(x_g^{t-1})$ ) then
15:      Aumente  $r_i$  e diminua  $A_i$ 
16:    end if
17:    Classifique os morcegos e encontre o melhor
18:  end for
19: end while

```

4. Experimentos Computacionais

Esta seção apresenta as funções de referência usadas para validar o algoritmo proposto. Também são apresentados a configuração dos valores atribuídos aos parâmetros. Finalmente, são discutidos as simulações e os resultados encontrados.

4.1. Funções de Referência e Configurações dos Parâmetros

É comum usar funções de referência com a hipótese de que a dificuldade apresentada por elas corresponde àquelas encontradas em aplicações reais. Assim, elas são frequentemente usadas para validar e comparar os algoritmos de otimização, bem como para validar novas abordagens de otimização global [Dieterich and Hartke 2012].

Para realizar os experimentos, foram escolhidas seis funções de referência que são aplicadas em problemas de minimização e que já foram utilizadas em vários estudos de BA [Fister Jr et al. 2013, Kabir and Alam 2014, Shan et al. 2016]. Para cada uma delas, a fórmula, o espaço de busca e a solução ideal são apresentados na Tabela 1. Esfera (f_1), Rosenbrock (f_2) e *Noisy Quartic* (f_3) são funções unimodais. Geralmente, elas são usadas para testar a capacidade do algoritmo em executar buscas locais. Por outro lado, Griewank (f_4), Rastrigin (f_5) e Ackley (f_6) têm vários mínimos locais e são usadas para verificar a habilidade do algoritmo para realizar buscas globais.

Os valores atribuídos aos parâmetros foram os mesmos usados por [Zhu et al. 2016]: sonoridade A_i e taxa de pulso r_i são fixados em 0.5, sonoridade

Tabela 1. Funções de Referência Avaliadas.

Fun	Fórmula	Espaço de Busca	Sol. Ótima
f_1	$f_1(x) = \sum_{i=1}^d x_i^2$	$-100 \leq x_i \leq 100$	$f(x^*) = 0$
f_2	$f_2(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-30 \leq x_i \leq 30$	$f(x^*) = 0$
f_3	$f_3(x) = \sum_{i=1}^d ix_i^4 + rand(0, 1)$	$-1.28 \leq x_i \leq 1.28$	$f(x^*) = 0$
f_4	$f_4(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 \leq x_i \leq 600$	$f(x^*) = 0$
f_5	$f_5(x) = \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i) + 10]$	$-5.12 \leq x_i \leq 5.12$	$f(x^*) = 0$
f_6	$f_6(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + \exp(1)$	$-32.76 \leq x_i \leq 32.76$	$f(x^*) = 0$

inicial A_0 é igual a 2 e sonoridade mínima A_{min} é igual a 0. Os valores atribuídos às constantes γ e α são 0.05 e 0.95, respectivamente.

4.2. Simulações e Resultados

Todas as rotinas foram implementadas em MATLAB R2013a. Os experimentos foram executados em um computador que usa um processador Intel Core i7, com 2,4 GHz de frequência, 8 GB de memória RAM e *Windows 10 Home Single Language* (64 bits).

Nos experimentos, são realizadas 30 execuções independentes. Em cada execução, são fixados o número de dimensões, o número de iterações e a quantidade de morcegos em 30, 900 e 50, respectivamente. Essa configuração também foi a mesma definida nos experimentos realizados em [Zhu et al. 2016]. Para avaliar os resultados da variante proposta, ela foi comparada com o BA original e com a variante da literatura *Modified Bat Algorithm* (MBA) [Yilmaz et al. 2014]. O critério usado para escolher a variante MBA como um parâmetro de avaliação da variante proposta, é o fato de MBA modificar equações que interferem no mecanismo de exploração do BA original.

As figuras 1, 2 e 3 comparam o processo de convergência médio do BA com o do BA modificado com o operador de mutação de Michalewicz (BA-Michalewicz), medido em 30 execuções independentes. O processo ocorre no espaço de busca das funções Esfera, Rosenbrock, *Noisy Quartic*, Griewank, Rastrigin e Ackley, respectivamente.

Nas subfiguras 1(a) e 1(b), que correspondem às funções Esfera e Rosenbrock, é observado que a velocidade de convergência da variante proposta, em relação ao BA original, é bastante superior. Na função Esfera, o BA original se mantém estagnado, praticamente, desde o início das primeiras iterações até o final. Isso mostra que ele convergiu prematuramente e a que a busca pela solução ótima global foi ineficaz. Isso também ocorre na função Rosenbrock. O BA original permanece estagnado, aproximadamente, a partir da iteração 60 até o final. Embora a velocidade de convergência da variante proposta, durante a otimização da função Esfera, seja reduzida a partir da iteração 360 e, para

a função Rosenbrock, a partir da 540, a nova variante encontra soluções muito melhores que as do algoritmo original.

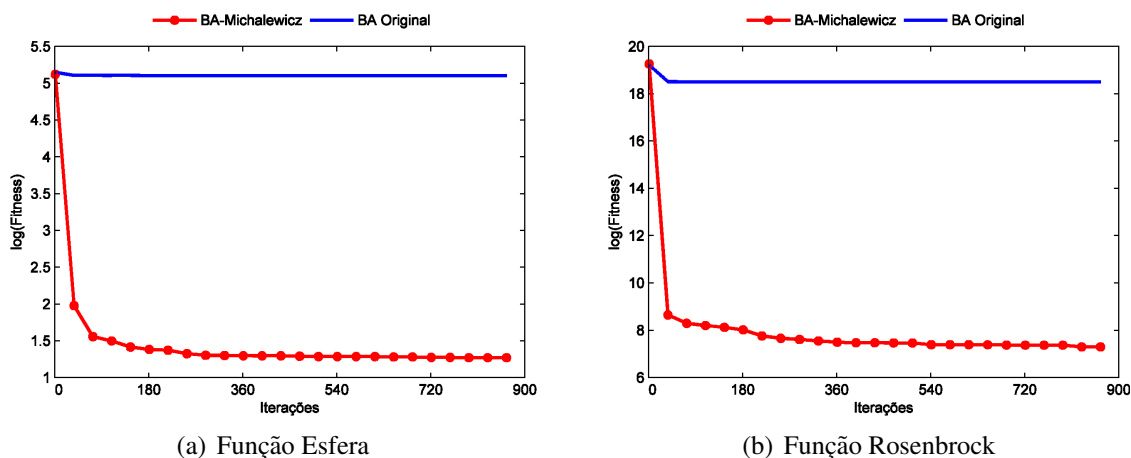


Figura 1. Processo de convergência médio nas funções Esfera e Rosenbrock.

Na função *Noisy Quartic* (subfigura 2(a)) e na função Griewank (subfigura 2(b)), os resultados da variante proposta também superam os do BA original. Na subfigura 2(a), o BA original estagna, praticamente, no início das iterações. E, na subfigura 2(b), ele inicia o seu processo de estagnação próximo da iteração de número 50. Durante todo o processo de busca, é observado que o algoritmo original tem uma baixa velocidade de convergência, apresentando poucas melhorias significativas em seus resultados. Por outro lado, a variante BA-Michalewicz apresenta um comportamento de convergência mais ativo e apresenta melhores soluções que o algoritmo original.

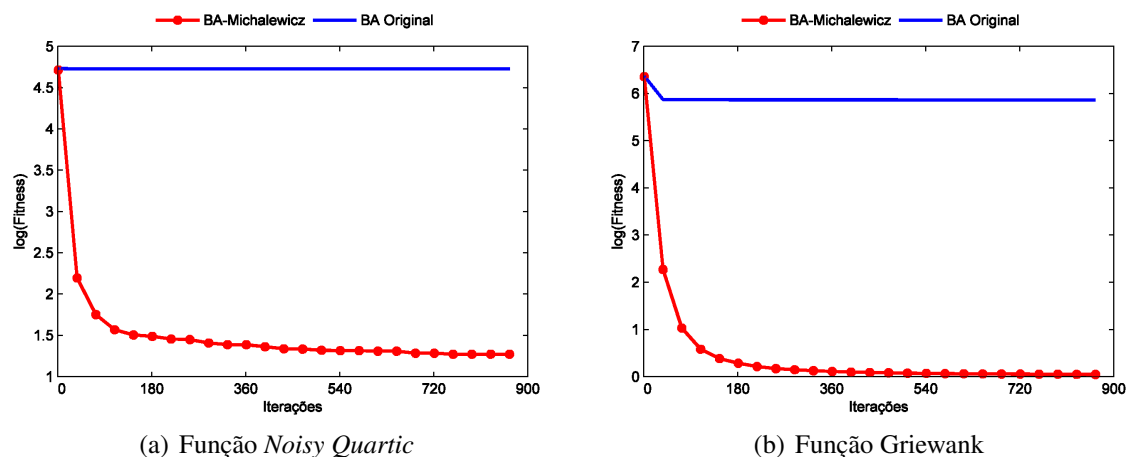


Figura 2. Processo de convergência médio nas funções *Noisy Quartic* e Griewank.

Na subfigura 3(a), que corresponde à função Rastrigin, o BA original estagna, mais uma vez, próximo das primeiras iterações e, na função Ackley (subfigura 3(b)), próxima da iteração 60. Além disso, o algoritmo original obtém soluções insatisfatórias quando comparado à variante proposta, a qual apresenta um comportamento de convergência bastante ativo durante o processo de otimização.

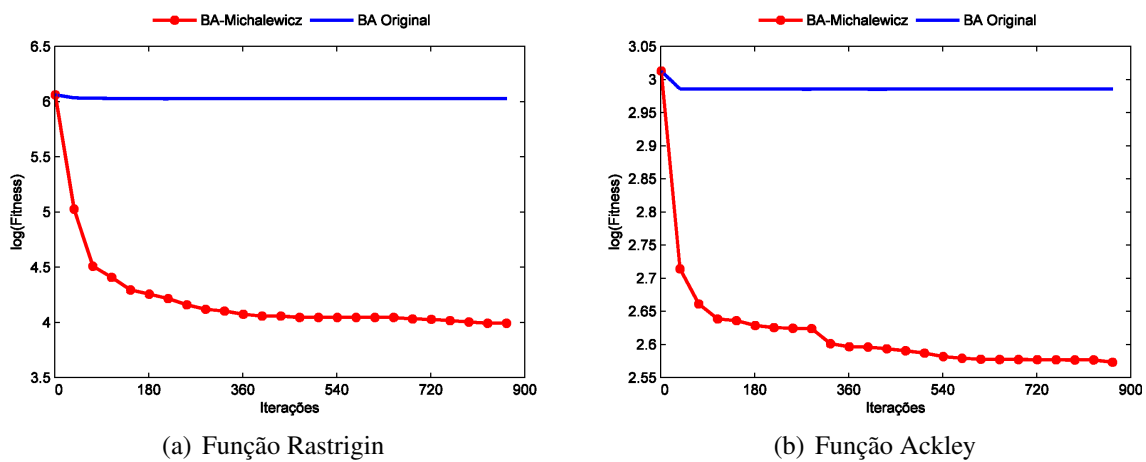


Figura 3. Processo de convergência médio nas funções Rastrigin e Ackley.

Na Tabela 2, são apresentados os resultados dos experimentos computacionais realizados. Em todas as funções avaliadas, os valores médios das soluções encontradas pela variante proposta são melhores que os do BA original e da variante MBA. Nota-se também que, em várias funções, a média da variante proposta é mais satisfatória que os melhores resultados do BA e da variante MBA, comprovando mais uma vez a superioridade do algoritmo proposto.

Em relação ao desvio padrão, que mede a dispersão em torno da média, a variante BA proposta obteve valores mais satisfatórios que o BA original e que o MBA em 66,37% dos experimentos. Apenas nos experimentos com as funções f_5 e f_6 (destacados com um *), o desvio padrão da variante proposta não foi melhor que o do MBA, embora o valor médio das soluções tenha sido. Isso comprova a estabilidade das soluções encontradas.

Tabela 2. Comparação entre BA original, MBA e o BA-Michalewicz.

Fun	BA original		MBA		BA-Michalewicz	
	Melhor	Média (Desvio)	Melhor	Média (Desvio)	Melhor	Média (Desvio)
f_1	1.14e+02	1.64e+02 (2.22e+01)	1.19e+01	2.30e+01 (5.95e+00)	7.36e-03	3.55e+00 (2.39e+00)
f_2	5.17e+07	1.08e+08 (3.61e+07)	1.11e+06	5.56e+06 (3.11e+06)	8.72e-02	1.42e+03 (1.55e+03)
f_3	5.90e+01	1.12e+02 (2.34e+01)	2.26e+00	4.69e+00 (1.71e+00)	4.06e-01	3.56e+00 (1.37e+00)
f_4	2.15e+02	3.51e+02 (9.23e+01)	4.89e+01	8.07e+01 (1.88e+01)	1.00e+00	1.04e+00 (1.83e-02)
f_5	3.47e+02	4.13e+02 (2.48e+01)	1.23e+02	1.54e+02 (1.68e+01)	1.84e-03	5.41e+01 (4.77e+01)*
f_6	1.77e+01	1.97e+01 (4.69 e-01)	1.19e+01	1.49e+01 (1.21e+00)	6.86e-02	1.30e+01 (7.77e+00)*

Wilcoxon é um teste estatístico não-paramétrico utilizado para comparar duas

amostras independentes. A hipótese nula H_0 afirma que as duas amostras vêm da mesma população, enquanto a hipótese alternativa H_1 afirma que uma delas possui valores maiores que a outra. Quando o p -value é menor que o nível de significância, decide-se rejeitar H_0 , isto é, há diferença significativa entre as amostras.

O Teste de Wilcoxon, com nível de significância de 0.05, é aplicado a fim de comparar os resultados das soluções encontradas pela variante proposta e pelo BA original. Ele é usado para verificar se os resultados da variante proposta são estatisticamente significativos quando comparados aos do BA original. Na otimização das funções f_1 , f_2 , f_3 , f_4 e f_5 o valor do p -value foi 1.86e-09 e, para a função f_6 , 4.65e-08, ou seja, inferior ao do nível de significância 0.05 estabelecido no teste. Assim, a hipótese H_0 foi rejeitada garantindo que não foram encontradas igualdades entre as soluções apresentadas pelos BA original e pela variante proposta neste trabalho.

Para cada uma das funções de referência avaliada, a Tabela 2 (melhores resultados são destacados em negrito) mostra a melhor solução encontrada, a média do valor de *fitness* e o desvio padrão do BA original, da variante MBA e da variante proposta, quando eles são executados 30 vezes. Os resultados mostram que a nova variante supera o BA original e a variante MBA. Ao comparar a variante proposta com o BA original e com o MBA, os resultados mostram sua superioridade no que diz respeito à estabilidade e à velocidade de convergência.

5. Conclusões

Este trabalho apresentou uma variante da meta-heurística inspirada no processo de ecolocalização dos morcegos conhecida como *Bat Algorithm* (BA). A nova variante consiste na combinação do BA original com o operador de mutação de Michalewicz. No entanto, este operador foi modificado para que, ao contrário da sua proposta original que altera apenas uma dimensão do vetor, todas as dimensões sejam alteradas. A proposta da variante é gerar diversidade para as soluções e reduzir os problemas causados pela convergência prematura, caracterizado pela estagnação do enxame em ótimos locais.

O desempenho do BA original, da variante MBA e da variante proposta foi comparado por meio de vários experimentos computacionais que foram realizados utilizando seis funções de referência: Esfera, Rosenbrock, *Noisy Quartic*, Griewank, Rastrigin e Ackley. Após os experimentos, observou-se que a nova variante apresentou um bom comportamento médio de convergência e obteve resultados superiores aos algoritmos citados no que diz respeito à qualidade e à estabilidade das soluções encontradas. Isso ocorreu porque o operador de mutação de Michalewicz modificado preservou a diversidade do algoritmo original, tornando o seu processo de convergência mais ativo.

Os resultados apontados neste artigo são promissores para o contexto das meta-heurísticas. Diante disso, o objetivo de trabalhos futuros é avaliar a adaptação de um conceito da área de Inteligência Computacional – que tem sido usado para aumentar a eficiência dos algoritmos meta-heurísticos – denominada *Elite Opposition-Based Learning* (EOBL) em uma outra meta-heurística bioinspirada conhecida como Algoritmo do Vaga-lume (*Firefly Algorithm* – FA). Assim como o Algoritmo do Morcego, FA também tem sido bastante utilizado para resolver problemas de otimização do mundo real.

Referências

- Chiaradonna, S., Di Giandomenico, F., and Murru, N. (2015). On enhancing efficiency and accuracy of particle swarm optimization algorithms. *International Journal of Innovative Computing, Information and Control (IJICIC)*, 11(4):1165–1190.
- Cordeiro, J. A., Parpinelli, R. S., and Lopes, H. S. (2012). Análise de sensibilidade dos parâmetros do bat algorithm e comparação de desempenho.
- Dieterich, J. M. and Hartke, B. (2012). Empirical review of standard benchmark functions using evolutionary global optimization. *arXiv preprint arXiv:1207.4318*.
- Dumitrescu, D., Lazzarini, B., Jain, L. C., and Dumitrescu, A. (2000). *Evolutionary computation*. CRC press.
- Evers, G. I. and Ghaliya, M. B. (2009). Regrouping particle swarm optimization: A new global optimization algorithm with improved performance consistency across benchmarks. In *SMC*, pages 3901–3908. IEEE.
- Fister Jr, I., Fister, D., and Yang, X.-S. (2013). A hybrid bat algorithm. *arXiv preprint arXiv:1303.6310*.
- Kabir, M. W. U. and Alam, M. S. (2014). Bat algorithm with self-adaptive mutation: a comparative study on numerical optimization problems. *International Journal of Computer Applications*, 100(10).
- Matos, S. A. and Carvalho, A. B. (2016). Traffic lights signaling optimization as a many-objective optimization problem. In *XIII Encontro Nacional de Inteligência Artificial e Computacional*, pages 421–432.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag.
- Mishra, S., Shaw, K., and Mishra, D. (2012). A new meta-heuristic bat inspired classification approach for microarray data. *Procedia Technology*, 4:802–806.
- Nakamura, R. Y., Pereira, L. A., Costa, K., Rodrigues, D., Papa, J. P., and Yang, X.-S. (2012). Bba: a binary bat algorithm for feature selection. In *25th Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 291–297.
- Reddy, V. U. and Manoj, A. (2012). Optimal capacitor placement for loss reduction in distribution systems using bat algorithm. *IOSR journal of Engineering*, 2(10):23–27.
- Shan, X., Liu, K., and Sun, P.-L. (2016). Modified bat algorithm based on lévy flight and opposition based learning. *Scientific Programming*, 2016.
- Silva, C. R. M. and Martins, S. R. (2013). An adaptive evolutionary algorithm for uwb microstrip antennas optimization using a machine learning technique. *Microwave and Optical Technology Letters*, 55(8):1864–1868.
- Storn, R. and Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359.
- Wang, G., Guo, L., Duan, H., Liu, L., and Wang, H. (2012). A bat algorithm with mutation for uav path planning. *The Scientific World Journal*, 2012.

- Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pages 65–74. Springer.
- Yilmaz, S. and Kucuksille, E. U. (2013). Improved bat algorithm (iba) on continuous optimization problems. *Lecture Notes on Software Engineering*, 1(3):279.
- Yilmaz, S., Kucuksille, E. U., and Cengiz, Y. (2014). Modified bat algorithm. *Elektronika ir Elektrotechnika*, 20(2):71–78.
- Zhang, J. W. and Wang, G. G. (2012). Image matching using a bat algorithm with mutation. In *Applied Mechanics and Materials*, volume 203, pages 88–93.
- Zhu, B., Zhu, W., Liu, Z., Duan, Q., and Cao, L. (2016). A novel quantum-behaved bat algorithm with mean best position directed for numerical optimization. *Computational intelligence and neuroscience*, 2016.

A Preference-guided Multiobjective Evolutionary Algorithm based on Decomposition

Daniel E. Souza¹, Fillipe Goulart¹, Lucas S. Batista², Felipe Campelo²

¹Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais, Brazil.

²Department of Electrical Engineering, Universidade Federal de Minas Gerais, Brazil.

danielpucplg@gmail.com, {fgsm-c, fcampelo, lusoba}@ufmg.br

Abstract. *Multiobjective evolutionary algorithms based on decomposition (MOEA/Ds) represent a class of widely employed problem solvers for multi-criteria optimization problems. In this work we investigate the adaptation of these methods for incorporating preference information prior to the optimization, so that the search process can be biased towards a Pareto-optimal region that better satisfies the aspirations of a decision-making entity. The incorporation of the Preference-based Adaptive Region-of-interest (PAR) framework into the MOEA/D requires only the modification of the reference points used within the scalarization function, which in principle allows a straightforward use in more sophisticated versions of the base algorithm. Experimental results using the UF benchmark set suggest gains in diversity within the region of interest, without significant losses in convergence.*

1. Introduction

Decomposition-based algorithms represent a widely used class of problem solvers for multiobjective optimization problems (MOPs). In particular, methods based on the Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) framework have been adopted for a variety of applications since its introduction in 2007 [Zhang and Li 2007], as evidenced by a recent comprehensive survey of the literature [Trivedi et al. 2016].¹

Similarly to dominance and indicator-based approaches, MOEA/Ds have been extensively developed and used in the context of *a posteriori* techniques to the solution of MOPs. This approach consists of first trying to obtain an extensive coverage of the Pareto-optimal front - which is generally seen as converging to a Pareto-optimal set of points homogeneously distributed in the space of objectives - and then delivering this set of options to a decision-making entity, which is tasked with the function of selecting a single solution for implementation.

Recent works have called this *a posteriori* approach into question, pointing out that in many cases one is not interested in actually mapping out the full Pareto front in detail, which can lead to wasteful optimization approaches that spend most of their computational budgets refining solutions that are clearly of no interest to the end user.

¹Throughout this text we use the term “MOEA/Ds” to refer to the class of algorithms based on decomposition for multiobjective optimization, i.e., the original MOEA/D and its many variants found in the literature.

Goulart and Campelo [Goulart and Campelo 2016] summarize this position (in the context of many-objective optimization), by declaring that “finding Pareto-optimal points does not necessarily mean that one has solved the practical many-objective optimization problem” [Goulart and Campelo 2016]. Indeed, even discounting the computational budget issue, it can be argued that returning hundreds of Pareto-optimal points for a decision-making entity to sort out and deal with poses an unnecessarily complex problem, which can be addressed by incorporating the preferences within the multiobjective optimization algorithm itself.

This work presents an adaptation of the approach originally proposed by Goulart and Campelo [Goulart and Campelo 2016] for generating preference-guided multiobjective evolutionary algorithms. In that work, the authors presented a framework for incorporating information regarding the stated preferences of a decision-making entity in the form of a reference point, but only in the context of dominance- and indicator-based algorithms. In the present work we extend those concepts to decomposition-based algorithms, which have been shown to perform well in the very classes of problems for which preference incorporation seems to be most critical [Goulart and Campelo 2016]: many-objective optimization problems [Asafuddoula et al. 2015, Li et al. 2015] and applied engineering problems (where preference information is often available) [Trivedi et al. 2016].

The remainder of this paper is organized as follows: Section 2 provides a quick introduction to the MOEA/D, and Section 3 reviews a few works dealing with preference incorporation within the MOEA/D structure. Section 4 discusses the concepts of the Preference-based Adaptive Region-of-interest (PAR) framework. The proposed PAR-MOEA/D is described in Section 5, and the results of computational experiments are presented in 6. Finally, Section 7 closes the paper with discussions, conclusions and perspectives.

2. Multiobjective Evolutionary Algorithms based on Decomposition

For the purposes of this work we deal with continuous MOPs subject only to box constraints, as defined as in (1):

$$\begin{aligned} \min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) &= (f_1(\mathbf{x}), \dots, f_{n_f}(\mathbf{x})) \\ \text{subject to: } \mathbf{x} &\in \mathbb{R}^{n_v} \mid x_i^{\min} \leq x_i \leq x_i^{\max}, \forall i \in \{1, \dots, n_v\} \end{aligned} \quad (1)$$

where n_f is the number of objectives, n_v is the number of decision variables, $\mathbf{x} \in \mathbb{R}^{n_v}$ represents a candidate solution, $\mathbf{f}(\cdot) : \mathbb{R}^{n_v} \mapsto \mathbb{R}^{n_f}$ is a vector of objective functions, and the feasible decision space (Ω) is defined by the real space \mathbb{R} bound by x_i^{\min} and x_i^{\max} , the lower and upper limits allowed for each variable. The image of the set Ω , $\mathbf{f}(\Omega)$, defines the set of attainable objective values [Miettinen 1999, Hidden and Hidden 2017].

MOEA/Ds decompose a MOP into a finite number of scalar optimization subproblems, each of which is defined by a weight vector and a scalar aggregation function. As discussed in [Hidden and Hidden 2017], a MOEA/D can be fully characterized by the design choices made for each of its components. For the algorithm considered in this work, these choices are:

- Decomposition strategy (which generates the weight vectors λ_i used for defining each subproblem): simplex-lattice design (SLD) [Chan 2000, Zhang and Li 2007];

- Scalar aggregation function (which provides the scalar subproblems to be minimized): as defined in Section 5;
- Neighborhood strategy (which regulates the locality of the exchange of information between subproblems): fixed neighborhood based on the distances between the weight vectors [Zhang and Li 2007];
- Objective function scaling: simple scaling [Hidden and Hidden 2017];
- Variation operators: differential mutation, binomial recombination, and polynomial mutation [Li and Zhang 2009];
- Update strategy (which selects which candidate solutions will compose the population at the end of each iteration): standard greedy selection [Zhang and Li 2007];
- Stop criterion: number of candidate solution evaluations.

The general structure of the MOEA/D is provided in Algorithm 1 [Hidden and Hidden 2017]. For more information on each module of the algorithm please refer to the references provided, particularly the original MOEA/D [Zhang and Li 2007] and the MOEA/D-DE [Li and Zhang 2009]. Details on the scalarizing functions will be given in Section 5.

Algorithm 1 MOEA/D structure

```

1:  $t \leftarrow 0$ 
2: Generate weights  $\lambda_i$ 
3: Generate initial population  $\mathbf{X}^{(t)}$ 
4: Define neighborhoods  $b_i$ 
5: while  $t \leq t_{max}$  do
6:    $\mathbf{X}'^{(t)} \leftarrow \text{DifferentialMutation}(\mathbf{X}^{(t)} \mid F)$ 
7:    $\mathbf{X}'^{(t)} \leftarrow \text{BinomialRecombination}(\mathbf{X}'^{(t)} \mid CR)$ 
8:    $\mathbf{X}'^{(t)} \leftarrow \text{PolynomialMutation}(\mathbf{X}'^{(t)} \mid \eta_M, p_M)$ 
9:   Evaluate solutions in  $\mathbf{X}^{(t)}$  and  $\mathbf{X}'^{(t)}$  using scalarizing function;
10:  Update population using standard greedy selection;
11:   $t \leftarrow t + 1$ 
12: end while

```

3. Preference-based MOEA/Ds

The incorporation of preference information into the structure of multiobjective evolutionary algorithms has been explored for some time (an extensive review of dominance- and indicator-based methods can be found in the references provided [Goulart and Campelo 2016, Goulart 2015]). In the present work three works incorporating preferences within the specific MOEA/D framework are briefly discussed.

The first reference dealing with this particular aspect in the context of decomposition-based approaches dates from 2011, in the form of a paper by Gong *et al.* [Gong et al. 2011]. That work proposes an interactive version of the MOEA/D, in which a human decision-maker (DM) is required, once every few iterations, to select its most preferred candidate solution from a subset of the current population. The main drawback of this approach is the need for iteratively querying the decision maker, which puts a heavy burden on the human operator. This can also make solutions subject to priming or anchoring effects [Tversky and Kahneman 1975], wherein the judgment of the DM

becomes subjectively biased by the solutions presented, and deviates from her rational utility model.

A second strategy was presented by Mohammadi, Omidvar and Li [Mohammadi et al. 2012], employing the concept of predefined reference points to bias the search towards template solutions provided by the DM. Their approach starts by running a common MOEA/D for a few iterations, and then transitions to a preference-guided scheme in which the weight vectors associated with the closest solution to each reference point are determined, and a new set of weight vectors is generated around each of these vectors. More recent versions of this method [Mohammadi et al. 2014] maintain the same general idea. While this eliminates the DM from the algorithmic loop, the method still relies on several additional external parameters that must be provided by the user - in general, it more than doubles the number of required user-defined parameters.

Finally, a third method was presented in 2015 by Pilát and Neruda [Pilát and Neruda 2015], which incorporates preferences via a coevolutionary approach in which the weights undergo an iterative refinement similar to that of the population. Preferences are initially expressed in the form of a binary judgment associated with each candidate solution as either preferred or non-preferred. The preference value associated to each point is then refined, and the preference function that emerges from this process is then used at each iteration as a quality value to compare the weight vectors, which are subject to a mutation-based variation. Despite presenting very interesting results for preference-guided multiobjective optimization, this approach also suffers from the same problem of an explosion in the number of free parameters. The algorithm structure also seems overly complex, which may possibly be an unavoidable price for incorporating arbitrarily complex preference structures within the framework of a multiobjective optimization approach, but can also be a result of the common practice of assembling an algorithm without much regard to the contribution of individual components, as suggested in the literature on component-based design and analysis of algorithms [Bezerra et al. 2016, Bezerra et al. 2015, Hidden and Hidden 2017].

Regardless, we argue that more parsimonious approaches are possible in the case of simple preference structures, which emerge in several engineering problems. The PAR framework [Goulart and Campelo 2016] represents a straightforward way to incorporate preferences within a MOEA, without the need for interactive querying or the addition of new parameters. While its current definition accepts only a single reference point as a template solution to express the preferences of the DM, its simplicity and easy incorporation into dominance- and indicator-based MOEAs motivated the current proposal of using it within the MOEA/D structure. The following section describes the adaptation of the MOEA/D for using the PAR framework for incorporating preferences into its optimization procedure.

4. The PAR framework

Assume the DM possesses aspiration levels z_i^r for the i -th objective, which collectively compose a *preference point*² $\mathbf{z}^r \in \mathbb{R}^{n_f}$. A straightforward way of satisfying the DM is

²The original term is *reference point* [Wierzbicki 1982]. However, we opted for *preference point* to prevent confusions with the actual reference point employed by the MOEA/D to generate the subproblems, and to emphasize its character of representing the DM's preferences.

by computing a solution that is closest to \mathbf{z}^r according to some indicator, as performed in techniques such as goal programming and weighted metrics [Miettinen 1999]. However, minimizing the distance to \mathbf{z}^r by means of a regular norm does not guarantee Pareto-optimality [Wierzbicki 1982], which motivated the use of *achievement scalarizing functions* (ASFs) [Wierzbicki 1982], such as the *augmented Tchebycheff function*:

$$s(\mathbf{f}'(\mathbf{x}), \mathbf{z}^{r'}) = \max_i \{f'_i(\mathbf{x}) - z_i^{r'}\} + \epsilon_a \sum_{i=1}^{n_f} (f'_i(\mathbf{x}) - z_i^{r'}) \quad (2)$$

where $\epsilon_a \in \mathbb{R}_{>0}$ is a small positive multiplier for the augmentation term. This formulation assumes that the objectives and the preference point are properly scaled [Goulart 2015], which can be guaranteed by defining:

$$\begin{aligned} f'_i(\mathbf{x}) &= \frac{f_i(\mathbf{x}) - f_{i,min}^{(t)}}{f_{i,max}^{(t)} - f_{i,min}^{(t)}}, \quad \forall i \in \{1, \dots, n_f\} \\ z_i^{r'} &= \frac{z_i^r - f_{i,min}^{(t)}}{f_{i,max}^{(t)} - f_{i,min}^{(t)}}, \quad \forall i \in \{1, \dots, n_f\} \end{aligned} \quad (3)$$

where $f_{i,min}^{(t)}$ and $f_{i,max}^{(t)}$ denote the smallest and largest values of the i th objective function at the iteration within which the evaluation is being performed, t .

If the underlying algorithm already possesses a mechanism to handle multiple solutions, as is the case in the MOEA/D, we can approximate a *region of interest* (ROI) that is close to the preference point. The underlying technique to define the ROI is dependent on each method. The Preference-guided Adaptive Region of interest (PAR) approach [Goulart and Campelo 2016] has the advantage of not requiring any additional parameters to be set by the user. Its general structure can be described as follows:

1. Given a set of available solutions $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_\mu\}$, compute the ASF value of each point with relation to the preference point \mathbf{z}^r using (2). Denote the smallest ASF value in the set as s_{\min} ;
2. Compute n_f auxiliary points by adding s_{\min} to each coordinate of \mathbf{z}^r :

$$\mathbf{z}_{aux,i}^r = \mathbf{z}^r + \mathbf{e}_i s_{\min}, \quad \forall i \in \{1, \dots, n_f\} \quad (4)$$

where $\mathbf{e}_i \in \mathbb{R}^{n_f}$ is the unit vector with the i -th coordinate set to unity and all others to zero.

3. Compute the ASF of each point of \mathbf{X} in relation to each $\mathbf{z}_{aux,i}^r$, and find the points with the smallest value for each auxiliary point. Denote these n_f points as $\mathbf{x}_{*,i}$;
4. The points belonging to the ROI are then defined as those for which $f_i(\mathbf{x}) \leq f_i(\mathbf{x}_{*,i})$ for any $i \in \{1, \dots, n_f\}$.

Notice that the size of the ROI is dependent only on the distance of the preference point \mathbf{z}^r to the image of the population in the space of objectives. This allows for an adaptive process that reduces the ROI size as the population converges, without having to adjust any additional parameters. More details are given in [Goulart and Campelo 2016].

5. PAR-MOEA/D

Incorporating preference information within the MOEA/D using the PAR framework is quite straightforward, requiring only a small modification in the way the scalarized value is calculated for each candidate solution. The procedure is contained within the evaluation step of the MOEA/D (Algorithm 1: line 9), and can be described as follows:

1. Follow the PAR structure to determine the points $\mathbf{x}_{*,i}$ (Section 4, step 3);
2. Based on the points $\mathbf{x}_{*,i}$, generate a new reference point $\hat{\mathbf{z}}$ with elements calculated as:

$$\hat{z}_j = \min_i f'_j(\mathbf{x}_{*,i}), \forall j \in \{1, \dots, n_f\} \quad (5)$$

where $f'_j(\cdot)$ is the value of the j -th objective function at the point, scaled according to (3).

3. For each point in the population, calculate the performance value using the MOEA/D scalarization strategy as defined below.

By following the procedure above, we are essentially adopting the idea of the PAR framework, namely that of adaptively modifying a region of interest, based on the stated preferences of the decision-making entity and the current state of the population. All aspects of the MOEA/D remain unchanged, with the exception of the scalarization procedure and reference point used therein. This means that, at least in principle, this adaptation can be easily adopted within existing implementations without much effort, regardless of specific variation operators, constraint handling approaches, decomposition methods, or any other aspects of the method.

5.1. Scalarization functions

In this work we investigate the use of several different scalarization functions within the proposed PAR-MOEA/D. Notice that in all cases the reference point used is $\hat{\mathbf{z}}$, calculated as in (5), and not the estimated ideal point used in regular, non-preference MOEA/Ds. In the definitions below, let \odot denote the Hadamard product; $\|\cdot\|_\infty$ the Tchebycheff norm; $\epsilon_0 \in \mathbb{R}_{>0}$ a small positive constant used to prevent divisions by zero; and $\epsilon_a \in \mathbb{R}_{>0}$ the small positive multiplier for the augmentation factor in the scalarization strategies that adapt the ASF.³

1. Weighted Tchebycheff (WT) [Miettinen 1999, Zhang and Li 2007]:

$$f^{agg}(\mathbf{x} \mid \boldsymbol{\lambda}, \hat{\mathbf{z}}) = \|\boldsymbol{\lambda} \odot (\mathbf{f}(\mathbf{x}) - \hat{\mathbf{z}})\|_\infty \quad (6)$$

2. Adjusted Weighted Tchebycheff (AWT) [Qi et al. 2014, Wang et al. 2013]:

$$f^{agg}(\mathbf{x} \mid \boldsymbol{\lambda}, \hat{\mathbf{z}}, \epsilon_0) = \|\boldsymbol{\rho} \odot (\mathbf{f}(\mathbf{x}) - \hat{\mathbf{z}})\|_\infty \quad (7)$$

where $\boldsymbol{\rho} \in \mathbb{R}^{n_f}$ has elements given as:

$$\rho_j = \frac{(\lambda_j + \epsilon_0)^{-1}}{\sum_{j=1}^{n_f} (\lambda_j + \epsilon_0)^{-1}}, \forall j \in \{1, \dots, n_f\}$$

³In this work we set $\epsilon_0 = \epsilon_a = 10^{-6}$.

3. WT with augmentation term from ASF (ASF-WT), an extension of the WT approach which incorporates the augmentation term from the ASF formulation in Equation (2):

$$f^{agg}(\mathbf{x} \mid \boldsymbol{\lambda}, \hat{\mathbf{z}}) = \|\boldsymbol{\lambda} \odot (\mathbf{f}(\mathbf{x}) - \hat{\mathbf{z}})\|_{\infty} + \epsilon_a \sum_{i=1}^{n_f} (f_i(\mathbf{x}) - \hat{z}_i) \quad (8)$$

4. AWT with augmentation term from ASF (ASF-AWT), an extension of the AWT approach which incorporates the augmentation term:

$$f^{agg}(\mathbf{x} \mid \boldsymbol{\lambda}, \hat{\mathbf{z}}) = \|\boldsymbol{\rho} \odot (\mathbf{f}(\mathbf{x}) - \hat{\mathbf{z}})\|_{\infty} + \epsilon_a \sum_{i=1}^{n_f} (f_i(\mathbf{x}) - \hat{z}_i) \quad (9)$$

It should be mentioned at this point that there is one existing method that employs scalarization by weighted ASF and a weight determination formula similar to the ASF-AWT in the context of preference-guided multiobjective optimization. This method, known as WASF-GA [Ruiz et al. 2014], defines a region of interest by means of a reference point, and employs a scalarization analogous to (9) to assign utility values to each point in a population. To our knowledge, there is no precedent for either ASF-WT or ASF-AWT in the MOEA/D literature, even though the use of analogous scalarization strategies in different contexts can be found in earlier works [Wierzbicki 1982, Miettinen 1999].

6. Experimental validation

For this preliminary testing of the PAR-MOEA/D, we employed all 2-objective problems of the UF benchmark set⁴, UF1-UF7, as test functions. The preference points were artificially created in the following way: given the true ideal point of a given test problem, $\tilde{\mathbf{z}}$, and a randomly selected efficient solution $\mathbf{f}(\mathbf{x}^*)$ of that problem, we generated the \mathbf{z}^r points over the line segment between these two points:

$$\mathbf{z}^r = \alpha \mathbf{f}(\mathbf{x}^*) + (1 - \alpha) \tilde{\mathbf{z}} - \epsilon_r \quad (10)$$

with $\alpha \in \{0, 0.25, 0.50, 0.75, 1\}$, to investigate the performance of the algorithms tested for preference points ranging between the ideal point and a region close to the true Pareto-optimal front. A small constant $\epsilon_r = 0.025$ is used to prevent the ROI from collapsing into a single solution when $\alpha = 1$ and \mathbf{z}^r lies exactly on the efficient front.

6.1. Algorithms

In this work we compare the PAR-MOEA/D variants described in the previous section against the two original PAR approaches, namely PAR-DEMO(nds) and PAR-DEMO(ind) [Goulart and Campelo 2016]. Both algorithms equip the PAR approach within the structure of the DEMO algorithm [Robič and Filipič 2005], the first (nds) using nondominated sorting as its main selection approach, and the second (ind) using the additive ϵ indicator instead.

Both PAR-DEMO approaches were configured as described in [Goulart and Campelo 2016], while the PAR-MOEA/D used the following configurations (arbitrarily set based on values commonly practiced in the MOEA/D literature, without any specific tuning effort):

⁴<http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm>

- Differential mutation: $F = 0.5$;
- Binomial recombination: $CR = 0.4$;
- Polynomial mutation: $\eta = 20$ and $p_m = 1/30$;
- Neighborhood size: $T = 20$;

All algorithms have a population size of $\mu = 100$. The stopping condition is set to interrupt the execution after 30,100 function evaluations, which corresponds to 300 iterations.⁵ All algorithms were run $n = 30$ times on each (problem, preference point) pair.

6.2. Quality indicators

The outcome of each algorithm was evaluated according to three quality indicators: convergence, pertinence and diversity [Goulart and Campelo 2016, Goulart 2015]. The specific definition of these three indicators is provided below.

For *convergence*, we employed the inverted generational distance (IGD) indicator [Zitzler et al. 2003], which is defined as follows: let \mathbf{X}_f denote the set of nondominated points returned at the end of a given run of an algorithm on a problem, and \mathbf{P}^* be a set of uniformly distributed points belonging to the Pareto-optimal front of the problem. Then:

$$IGD(\mathbf{X}, \mathbf{P}^*) = \frac{\sum_{\forall \mathbf{p}^* \in \mathbf{P}^*} \left[\min_{\mathbf{x}_i \in \mathbf{X}_f} \{ \|\mathbf{p}^* - \mathbf{x}_i\|_2 \} \right]}{|\mathbf{P}^*|} \quad (11)$$

Given that the aim of the algorithms under comparison is to approximate only points within their ROI, only Pareto-optimal points belonging to the exact ROI of each method (which can be easily obtained by following the procedure from Section 4 using the full set \mathbf{P}^*) were used for calculating this indicator.

For *diversity*, we used the Hierarchical Cluster Count (HCC) indicator, as proposed in [Guimarães et al. 2009]. This indicator, which is able to measure both the spread and the uniformity of a set of points and does not require any external parameter, is calculated as follows:

1. Set $k = 0$, $\tau_k = 0$ and $r_k = 0$;
2. Let each point $\mathbf{x}_i \in \mathbf{X}_f$ define a cluster, C_i , with itself as the single element;
3. Find the clusters C_{i1} and C_{i2} with the smallest value of separation distance, using complete linkage [Guimarães et al. 2009]. Assign this minimal value of separation distance to r_{k+1} and join these clusters;
4. Set $\tau_{k+1} = \tau_k + (r_{k+1} - r_k)(|\mathcal{P}| - k + 2)$ and $k \leftarrow k + 1$;
5. Go back to step 3 until there is only one cluster left. Return τ_k as a value of the diversity of the set \mathbf{X}_f .

For the calculation of this indicator, the final population of each algorithm was standardized to the interval $[0, 1]^{n_f}$ in the *objective space*, and the diversity was computed for these transformed data.

Finally, the *pertinence* indicator, which measures the degree of satisfaction of the preferences expressed by the DM, was calculated as the smallest ASF in relation to the

⁵The initial population requires μ evaluations before the iterations start to be counted.

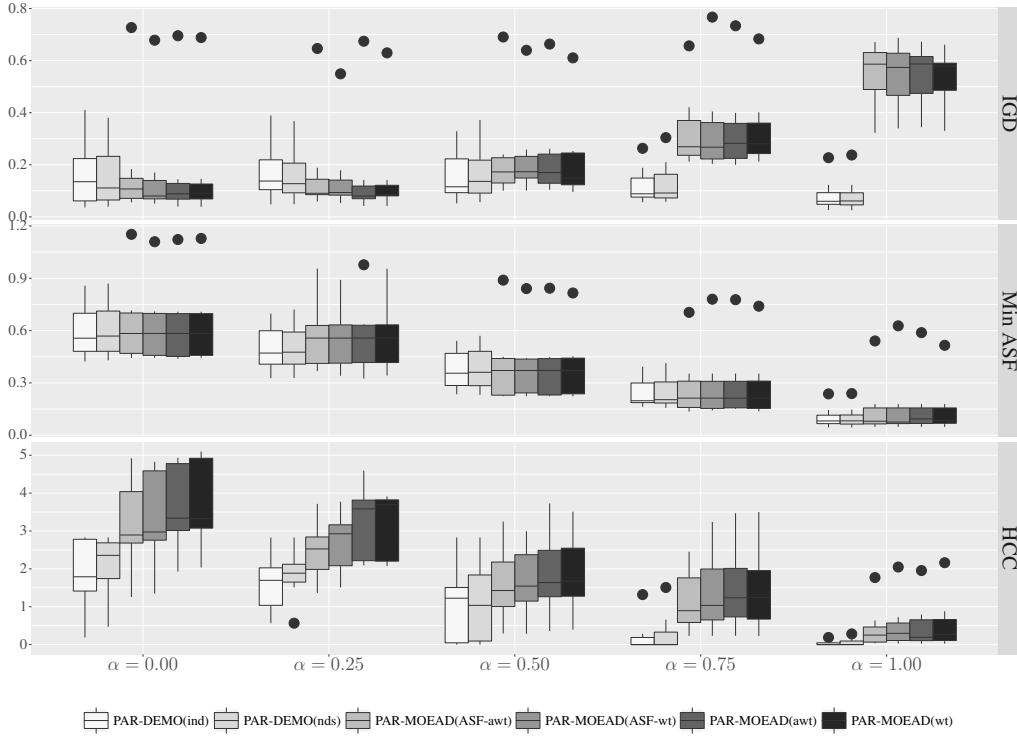


Figure 1. Box plots of the performance of each algorithm for each value of preference point distance, regarding the three quality indicators. Each box is composed by the averages of the performance of an algorithm on all problems for that value of α .

preference point \mathbf{z}^r , as suggested by [Goulart and Campelo 2016]:

$$s_{min}(\mathbf{X}_f | \mathbf{z}^r) = \min_{\mathbf{x}_i \in \mathbf{X}_f} \{s(\mathbf{f}(\mathbf{x}_i), \mathbf{z}^r)\} \quad (12)$$

The scaling when computing this indicator was performed as in (3), but employing the true ideal and Nadir solutions of each test problem.

Before proceeding, it is important to highlight that IGD and pertinence are indicators for which *smaller is better*, whereas for the HCC *larger is better*.

6.3. Results

The results obtained for the 30 replicates, summarized by means within each (problem instance, preference point distance, indicator) tuple, are illustrated in Figure 1. From the box plots, a few trends seem to be apparent, as discussed below.

First, IGD values seem to be slightly better for the PAR-MOEAD/D approaches in lower values of α (i.e., for preference points farther from the true Pareto-optimal front), with comparable average values but smaller variability. These approaches start losing to the PAR-DEMO methods as the preference points approach the attainable region of the space of objectives.

With respect to diversity, the PAR-MOEAD/D approaches appear to return superior (i.e., larger) values of the HCC indicator across all values of α , when compared to the

PAR-DEMO ones. The absolute HCC values of all methods is reduced as α increases, which is consistent it being harder to maintain good diversity in a smaller ROI.

Finally, regarding the pertinence indicator, which measures the degree of satisfaction of the stated preferences, the results also seem to be comparable across all values of α , with the PAR-DEMO approaches seemingly slightly better for preference points closer to the true Pareto-optimal front.

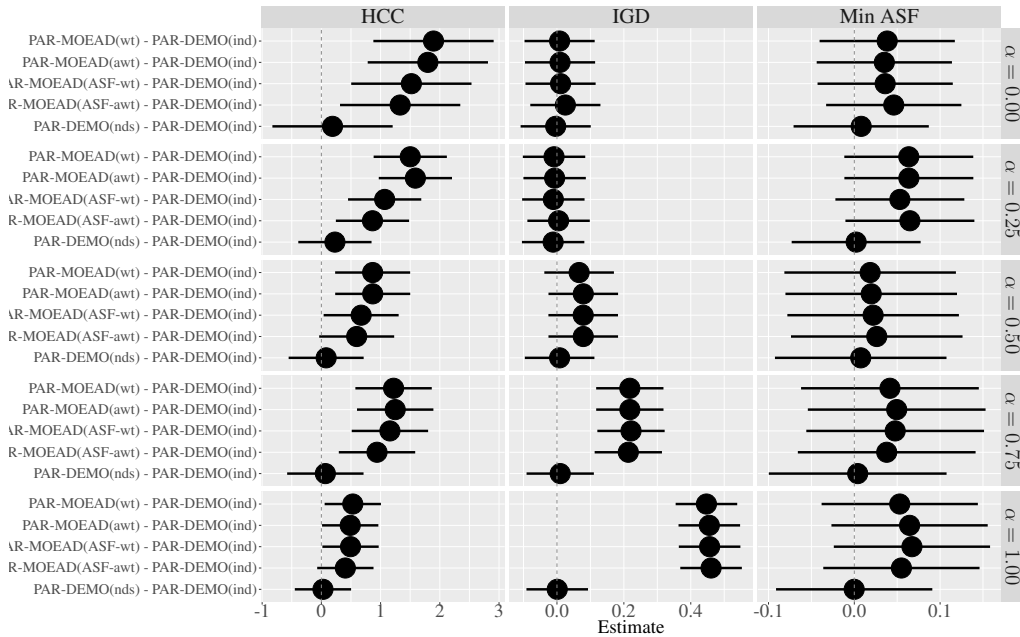


Figure 2. Simultaneous 95% confidence intervals for differences between each algorithm and PAR-DEMO(ind) in each quality indicator.

To reinforce these qualitative considerations, an RCBD ANOVA model [Montgomery 2012] was fit for the data within each value of α , with *algorithm* as the experimental factor and *problem instance* as a blocking variable. After fitting the model, the PAR-DEMO(ind) was selected as a reference method (since its performance is similar to the other PAR-DEMO approach for all cases) and all-vs-one paired comparisons were performed using Dunnett contrasts [Crawley 2007]. The results of these comparisons are illustrated in the form of simultaneous 95% confidence intervals in Figure 2, and corroborate the observations derived from examining the box plots.

7. Conclusion

In this work the Preference-guided Adaptive Region-of-interest (PAR) framework was included into the MOEA/D structure, allowing this method to incorporate preference information in its search procedure. While only a simple MOEA/D was used as the base algorithm in this work, the introduction of the PAR approach is straightforward, requiring only a modification in the procedure for calculating the scalarized utility of each point, which allows for its use with more sophisticated MOEA/D variants without much effort.

Results obtained using the UF benchmark set for different preference point specifications suggest that the PAR-MOEA/D versions tend to compare favorably to the PAR-DEMO approaches in terms of diversity of the nondominated set returned, while main-

taining the same overall convergence and pertinence characteristics. Since the original PAR work [Goulart and Campelo 2016] already shows that the PAR-DEMO approaches used here as the comparison baseline tend to outperform other preference-based MOEAs (such as the R-DEMO and PBEA) in terms of both convergence and pertinence, but lose in terms of diversity, the use of MOEA/D as the basis for the PAR framework (which improves diversity over the PAR-DEMO) seems like an interesting step in the direction of improving the performance of PAR-based methods in the treatment of multiobjective optimization problems for which preference information is available.

Future works include the investigation of more sophisticated MOEA/D variants as base algorithms for the PAR framework, the incorporation of methods for treating nonlinear constraints, and tests in many-objective scenarios.

Acknowledgment

This work has been partially funded by XXXXXXXXXXXXXXXX.

References

- Asafuddoula, M., Ray, T., and Sarker, R. (2015). A decomposition based evolutionary algorithm for many objective optimization. *IEEE Trans. Evolutionary Computation*, 19(3):445–460.
- Bezerra, L., ez, M. L.-I., and Stützle, T. (2016). Automatic component-wise design of multi-objective evolutionary algorithms. *IEEE Trans. Evolutionary Computation*, 20(3):403–417.
- Bezerra, L. C. T., López-Ibáñez, M., and Stützle, T. (2015). To DE or not to DE? multi-objective differential evolution revisited from a component-wise perspective. In *Lecture Notes in Computer Science*, pages 48–63. Springer Nature.
- Chan, L. Y. (2000). Optimal designs for experiments with mixtures: A survey. *Communications in Statistics: Theory and Methods*, 29(9–10):2281–2312.
- Crawley, M. J. (2007). *The R Book*. Wiley Publishing, 1st edition.
- Gong, M., Liu, F., Zhang, W., Jiao, L., and Zhang, Q. (2011). Interactive MOEA/D for multi-objective decision making. In *Proc. Genetic and Evolutionary Computation Conference (GECCO)*, pages 721–728, New York, NY, USA. ACM.
- Goulart, F. (2015). Preference-guided evolutionary algorithms for optimization with many objectives. Master’s thesis, Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil.
- Goulart, F. and Campelo, F. (2016). Preference-guided evolutionary algorithms for many-objective optimization. *Information Sciences*, 329:236–255.
- Guimarães, F. G., Wanner, E. F., and Takahashi, R. H. (2009). A quality metric for multi-objective optimization based on Hierarchical Clustering Techniques. *2009 IEEE Congress on Evolutionary Computation*, pages 3292–3299.
- Hidden, A. and Hidden, A. (2017). [title hidden for blind peer review]. [*Under review*].
- Li, H. and Zhang, Q. (2009). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evolutionary Computation*, 13(2):284–302.

- Li, K., Deb, K., Zhang, Q., and Kwong, S. (2015). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans. Evolutionary Computation*, 19(5):694–716.
- Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston.
- Mohammadi, A., Omidvar, M., and Li, X. (2012). Reference point based multi-objective optimization through decomposition. In *Proc. IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.
- Mohammadi, A., Omidvar, M. N., Li, X., and Deb, K. (2014). Integrating user preferences and decomposition methods for many-objective optimization. In *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE.
- Montgomery, D. (2012). *Design and Analysis of Experiments, 8th Edition*. John Wiley & Sons, Incorporated.
- Pilát, M. and Neruda, R. (2015). Incorporating user preferences in MOEA/D through the coevolution of weights. In *Proc. Genetic and Evolutionary Computation Conference (GECCO)*, pages 727–734, Madrid, Spain.
- Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., and Wu, J. (2014). MOEA/D with adaptive weight adjustment. *Evolutionary Computation*, 22(2):231–264.
- Robič, T. and Filipič, B. (2005). Demo: Differential evolution for multiobjective optimization. In *Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005*. Springer Berlin Heidelberg.
- Ruiz, A. B., Saborido, R., and Luque, M. (2014). A preference-based evolutionary algorithm for multiobjective optimization: the weighting achievement scalarizing function genetic algorithm. *Journal of Global Optimization*, 62(1):101–129.
- Trivedi, A., Srinivasan, D., Sanyal, K., and Ghosh, A. (2016). A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Trans. Evolutionary Computation*, PP(99):1–23.
- Tversky, A. and Kahneman, D. (1975). Judgment under uncertainty: Heuristics and biases. In *Utility, Probability, and Human Decision Making*, pages 141–162. Springer Netherlands.
- Wang, R., Zhang, T., and Guo, B. (2013). An enhanced MOEA/D using uniform directions and a pre-organization procedure. In *Proc. IEEE Congress on Evolutionary Computation*, pages 2390–2397, Cancún, Mexico.
- Wierzbicki, A. P. (1982). A mathematical basis for satisficing decision making. *Mathematical Modelling*, 3(5):391–405.
- Zhang, Q. and Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evolutionary Computation*, 11(6):712–731.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., and Fonseca, V. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evolutionary Computation*, 7(2):117–132.

Simulação Multiagentes: Áreas Atingidas por Terremotos com Focos de Incêndio

Ana Júlia Fontoura¹, Rejane Frozza^{1,2}, Daniela Bagatini^{1,3}, Gilberto Dessbesell¹, Tauame Aguiar Pacce¹

¹Departamento de Computação

²Programa de Pós-Graduação em Sistemas e Processos Industriais
Universidade de Santa Cruz do Sul (UNISC)

Av. Independência, 2293 – Bairro Universitário
Santa Cruz do Sul – RS - 96815-900

³Programa de Pós-Graduação em Informática na Educação (PGIE - UFRGS)

ana.fontoura@hotmail.com, [frozza, bagatini]@unisc.br,
gdbeto@gmail.com, tauamepacce@hotmail.com

Abstract. *The scenario created by an earthquake presents many challenges for rescue teams, such as landslides, disruptions in the supply of water and energy, and others, may occur. This paper presents a development proposal for a Multiagent System composed of intelligent agents able to fight multiple fires in an urban environment simulating a post-earthquake situation. The environment is explored by dividing the map into sectors, through K-means clustering, and the decision of which fire each agent will combat is based on ANN (Artificial Neural Networks). During evaluation, the proposed strategy was shown to be effective in environment exploration and on which fires should be fought.*

Resumo. *O cenário criado por um terremoto apresenta muitos desafios para equipes de resgate, uma vez que desabamentos, interrupções no fornecimento de água e energia, incêndios, entre outros, podem ocorrer. Assim, este trabalho descreve um Sistema Multiagentes composto por agentes inteligentes que devem combater múltiplos incêndios em um ambiente urbano simulando uma situação pós-terremoto. O ambiente é explorado dividindo o mapa em setores, através da técnica de Clusterização, com o algoritmo K-means, e a decisão de qual incêndio cada agente combaterá é baseada em RNA (Redes Neurais Artificiais). Durante a validação, a estratégia proposta se mostrou efetiva na exploração do ambiente e na seleção de qual incêndio combater.*

1. Introdução

Desde o início do século XXI, mais de um milhão de pessoas já morreram em decorrência de desastres naturais, sendo que destas, mais de 700 mil foram vítimas de terremotos. Além disto, conforme Guha-Supir, Below e Hoyois (2015), terremotos são a segunda maior causa de desabrigados e feridos, apesar de não fazerem parte dos desastres mais frequentes.

O resgate de vítimas em desastres naturais é um problema social de alta relevância. Segundo Singh, Modanwal e Basak (2011), as equipes de resgate agem frequentemente em ambientes perigosos e cheios de incertezas. Por meio da implementação de agentes inteligentes, é possível minimizar a utilização de humanos e animais em equipes de combate a desastres naturais. Esta redução é ainda mais importante no caso de terremotos, onde frequentemente as estruturas das edificações estão abaladas, impondo um risco iminente à vida de todos os envolvidos na ação de resgate.

Além da preservação das vidas envolvidas nos esforços de salvamento, a utilização de agentes inteligentes permite aumentar a eficiência das equipes. Pois, segundo Kitano e Tadokoro (2001), a magnitude de um desastre, e a velocidade na qual o sistema precisa mudar de acordo com a situação, está muito além da capacidade humana de planejamento em um ambiente extremamente hostil. Portanto, pode-se reduzir o tempo de salvamento, e conseqüentemente, o número de vítimas fatais.

O objetivo principal é desenvolver uma estratégia de aprendizagem, baseada em redes neurais artificiais e clusterização, em agentes inteligentes para atuação em situações de calamidade, com foco em áreas atingidas por terremotos. Para a validação desta estratégia, foi desenvolvido um simulador próprio para analisar o comportamento dos agentes em um cenário pós terremoto com focos de incêndio. A rede neural artificial foi usada para auxiliar a tomada de decisão do agente bombeiro e a clusterização para a divisão do terreno/mapa, reduzindo o escopo de atuação dos agentes.

A fim de se obter resultados relevantes na área de resgate em desastres naturais, este trabalho utiliza conceitos de aprendizagem em agentes, aplicando-os em um ambiente representando uma situação pós-terremoto. Com isto, pretende-se aumentar a eficiência das equipes de resgate, além de expor menos vidas aos riscos impostos pelo ambiente. Para tanto, são estudadas as Redes Neurais Artificiais para serem utilizadas como estratégia de aprendizado para a ação dos agentes no cenário de terremoto, já que são baseadas em aprendizado supervisionado e não supervisionado.

Segundo Prado *et al.* (1993), um sistema de Inteligência Artificial Distribuída (IAD) pode ser visto como um grupo de entidades inteligentes, chamadas de agentes, que interagem entre si. Ribeiro (2001) afirma ainda que uma das subáreas da IAD, a de Sistemas Multiagentes, representa uma forma de analisar, projetar e implementar sistemas de software complexos, sendo alvo de intensa atividade de investigação e desenvolvimento. Este tipo de tecnologia tem sido utilizado em variadas aplicações como, por exemplo, comércio eletrônico, gerência de redes, interfaces adaptáveis, robótica, resposta ao desastre e sistemas de controle de tráfego.

Segundo Russel e Norvig (2010), o primeiro modelo matemático de um neurônio foi proposto por McCulloch e Pitts em 1943. Este modelo de neurônio é ainda utilizado no estudo de Redes Neurais Artificiais. Uma RNA é formada quando múltiplos neurônios são conectados. Os neurônios são conectados em camadas, e uma RNA deve possuir no mínimo 2 camadas, uma para entrada de dados e outra para saída. Porém uma rede pode possuir qualquer número de camadas entre as de entrada e saída, sendo estas chamadas de camadas escondidas.

A prática de classificar objetos de acordo com suas similaridades, conforme Jain e Dubes (1988), é a base de grande parte dos trabalhos científicos, e além disso, a organização dos dados em grupos é uma das formas fundamentais de entendimento e aprendizagem. As técnicas de clusterização têm como objetivo buscar padrões em dados desordenados, classificando e organizando os objetos de acordo com suas semelhanças, distância ou densidade em um espaço multidimensional (JAIN e DUBES, 1988). Segundo Sanches (2003), um dos algoritmos mais frequentemente utilizados para clusterização é o K-means, proposto por James MacQueen em 1967, onde o usuário/especialista é responsável por definir o número de partições, ou clusters, a serem encontradas.

O artigo está organizado nas seguintes seções: A seção 2 apresenta a descrição dos trabalhos relacionados ao tema desta pesquisa; a seção 3 descreve a metodologia e o sistema (simulador multiagentes) desenvolvido; na seção 4 são apresentados os resultados atingidos e a seção 5 apresenta a conclusão.

2. Trabalhos Relacionados

Nesta seção, é apresentada uma análise comparativa das principais características de alguns trabalhos selecionados, dedicados ao estudo de Sistemas Multiagentes em combate a desastres naturais.

O trabalho de Liang, Yang e Deng (2014) propôs a utilização da técnica de clusterização com o algoritmo *K-means* para dividir áreas do mapa (relativo aos locais atingidos) entre grupos de agentes do tipo *Police Force* na *Robocup Rescue*.

Já no trabalho desenvolvido por Pereira, Nardin e Sichman (2011) foi adotada uma abordagem de coordenação dos agentes da *Robocup Rescue* de maneira que o maior número de vítimas seja salva e o menor número de construções seja consumida por incêndios. O trabalho propõe uma abordagem de alocação de tarefas baseada em Planejamento Global Parcial para coordenar agentes na *Robocup Rescue*.

Em Khouj *et al.* (2011), o trabalho tem como objetivo criar e treinar previamente um agente capaz de agir em cenários de emergência, medindo a capacidade operacional e disponibilidade de recursos nas infraestruturas mais críticas e redirecionando os recursos. O foco de aplicação do trabalho está em gerenciar os recursos disponíveis em uma situação de desastre, a fim de aumentar o número de vítimas do desastre que receberão alta nos hospitais ou postos de atendimento.

O trabalho proposto por Ramchurn *et al.* (2015) é um sistema que coleta informações sobre desastres através de redes sociais, organizações confiáveis, e veículos aéreos não tripulados (*drones*), e realiza uma avaliação destas informações, a fim de auxiliar humanos, com foco na recuperação de áreas atingidas por desastres. É demonstrada, então, a integração de agentes e humanos em equipes chamadas de *Human-Agent-Collectives*. No ano seguinte, Ramchurn *et al.* (2016) analisa um ambiente de desastre como incerto e dinâmico, já que novas tarefas e perigos podem aparecer no local. O trabalho desenvolvido refere-se ao uso de técnicas de decisão para solucionar problemas de alocação de tarefas por meio de planejamento baseado em agentes, no qual os humanos são guiados pelos agentes na execução das tarefas.

No trabalho de Barroso *et al.* (2013), são demonstradas as estratégias da equipe *LTI Agent Rescue* para a competição *Robocup Rescue* de 2013. A coordenação dos agentes adotada é completamente distribuída, não havendo utilização dos agentes centrais, para evitar o ponto único de falha e atraso na propagação de informações, decorrentes da utilização deste tipo de agente. Para a exploração, os autores adotam uma estratégia de particionamento simples, sem levar em consideração a distribuição dos prédios, dividindo o mapa em n setores iguais, onde n é o número de agentes policiais.

O trabalho de Pessin *et al.* (2007) propõe um Sistema Multiagente heterogêneo com o objetivo de combater incêndios florestais sem o envolvimento de humanos. Na comunicação dos agentes é utilizado um sistema de Quadro-negro, onde as mensagens vão do agente monitor para o agente líder, e do agente líder para os agentes de combate.

O trabalho de Pessin, Osório e Musse (2008) tem como objetivo projetar um Sistema Multiagente, formado por robôs controlados, agentes inteligentes que utilizam Redes Neurais Artificiais e Algoritmos Genéticos para determinar a melhor forma de combater incêndios florestais através da técnica de combate indireto por abertura de aceiros.

Quase todos os trabalhos implementados para a *Robocup Rescue* utilizaram técnicas de particionamento do mapa para a exploração do cenário. Isso demonstra que uma abordagem válida para resolver a exploração do problema de resgate é o particionamento do mapa. Dentre os trabalhos que utilizaram o simulador da *Robocup Rescue*, o aprendizado não foi utilizado no processo de tomada de decisão das tarefas de resgate. Apesar dos trabalhos que não utilizaram o simulador apresentarem diferentes métodos de aprendizado, os trabalhos de Ramchurn *et al.* (2015) e Ramchurn *et al.* (2016) abrem espaço para o envolvimento de humanos nas tarefas de resgate. No trabalho de Barroso *et al.* (2013) destaca-se como interessante a avaliação dos incêndios no ambiente através de uma heurística, e no trabalho de Liang, Yang e Deng (2014) destaca-se a utilização do método de clusterização com o algoritmo *K-means* para particionar o mapa em regiões de maneira mais igualitária em relação ao método criado por Barroso *et al.* (2013). No trabalho de Pessin *et al.* (2007) existe um maior nível de abstração no uso da RNA, uma vez que é utilizada para indicar a direção que o agente deve se mover para evitar a vegetação mais densa. No trabalho de Pessin, Osório e Musse (2008) foi reduzido o nível de abstração, atribuindo à RNA a tarefa de acionar os motores do agente diretamente, ao invés de apenas indicar a melhor direção.

3. Metodologia

Nesta seção é apresentada a metodologia e a descrição de todas as etapas do desenvolvimento: Modelo de agente bombeiro inteligente, baseado nos trabalhos de Barroso *et al.* (2013) e Liang, Yang e Deng (2014), para combate a incêndios, e do simulador de situação pós-terremoto onde a atuação dos agentes é validada. Para dar suporte ao agente bombeiro na identificação de focos de incêndio, também foi desenvolvido um agente explorador. São apresentadas as estratégias baseadas em redes neurais artificiais, clusterização e o cenário de atuação dos agentes.

3.1. Arquitetura do sistema multiagentes

A arquitetura do sistema multiagentes se divide em duas partes: Os agentes e o simulador. O simulador é composto por três módulos: Gerador de Incêndios, Mapa e *Core*, enquanto os agentes são divididos em Bombeiros e Exploradores, cada um com sua própria arquitetura. Uma visão geral da arquitetura e do fluxo de informação entre seus componentes pode ser observada na Figura 1.

O tempo na simulação é controlado pelo *Core*, e é dividido em intervalos discretos, ou ciclos. Em cada ciclo são realizados cálculos por parte do gerador de incêndio para determinar as mudanças que ocorrem no ambiente, e os agentes determinam suas ações, a fim de mitigar o dano causado pelos incêndios. Os agentes interagem com o *Core* para realizar suas ações.

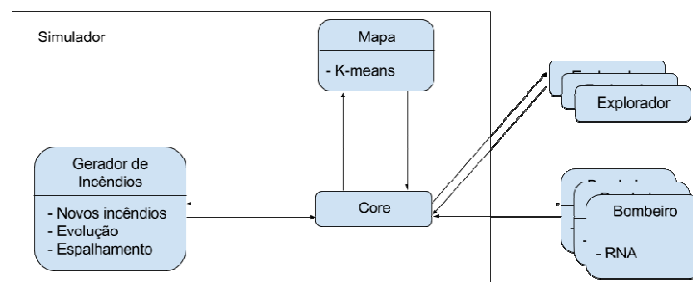


Figura 1. Arquitetura do sistema multiagentes

3.1.1 Mapa

O módulo Mapa é uma estrutura de dados baseada em matriz, onde cada célula armazena diversos objetos com seus respectivos atributos: Incêndios, Agentes bombeiro, Agentes explorador, Prédios, Ruas e Hidrantes.

Cada prédio do mapa pode ser feito de um dos seguintes materiais: Alvenaria, Alvenaria e Madeira, ou Madeira. Cada um destes materiais tem impactos diferentes na evolução e espalhamento dos incêndios, conforme será explicado na próxima subseção. Além de armazenar estas informações, o Mapa também é responsável por distribuir os prédios em setores através da clusterização com o algoritmo *K-means*. Esta divisão tem o objetivo de tornar mais eficiente a exploração do mapa, por parte dos agentes.

Para cada setor é gerado um ponto aleatório no mapa, chamado de centroide. Depois, cada prédio do mapa é atribuído ao setor cujo centroide é o mais próximo de si. Depois que todos os prédios são atribuídos a um setor, a posição de cada centroide é recalculada tomando a média de todas as posições dos prédios de seu setor. Se qualquer centroide mudar de lugar, o processo de atribuir os prédios aos setores do centroide mais próximo é repetido. A repetição continua até que os centroides parem de mudar de lugar, ou até que o tempo limite de 5 segundos seja atingido.

3.1.2 Gerador de Incêndios

O Gerador de Incêndios é responsável por controlar a evolução, espalhamento e surgimento dos incêndios da simulação. Ele também calcula, a cada ciclo da simulação,

a área que deve ser queimada em cada prédio com incêndio. O funcionamento geral deste módulo pode ser observado na Figura 2.

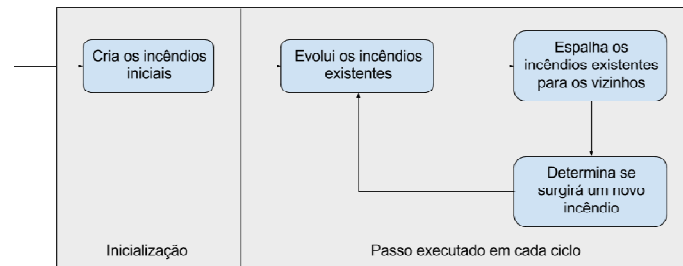


Figura 2. Funcionamento do Gerador de Incêndios

Cada incêndio possui, além de sua localização, uma intensidade e energia: A intensidade influencia o dano que o incêndio causa no prédio a cada ciclo. Esta é a única informação do incêndio visível aos agentes da simulação, e é um valor inteiro podendo variar de 0 (Extremamente Baixa) até 6 (Extremamente Alta). A energia indica o quão difícil será apagar o incêndio. No primeiro ciclo do surgimento de um incêndio, ela é determinada pela intensidade inicial. Nos ciclos posteriores ao do surgimento, na etapa de evolução, é a energia que é incrementada, e posteriormente baseada nessa energia, a intensidade é recalculada. A energia permite que um incêndio evolua sua intensidade de forma suave a cada ciclo, e também possibilita realizar ajustes mais finos na velocidade que os bombeiros apagam os incêndios, em comparação com utilizar apenas a intensidade. Quando um incêndio atinge a intensidade máxima (Extremamente Alta), a energia para de crescer.

Existem quatro fatores que podem ser definidos pelo usuário na interface do sistema: Fator de evolução (velocidade que a energia do incêndio cresce), fator de espalhamento (afeta a chance de um incêndio se espalhar para prédios vizinhos), fator de surgimento (chance de surgir um novo incêndio em local aleatório no mapa) e fator de velocidade de queima (dano causado a cada ciclo pelo incêndio).

Além dos fatores, o material dos prédios também influencia o comportamento dos incêndios. Cada material possui um coeficiente de inflamabilidade c_i , que é um multiplicador utilizado nos cálculos de crescimento e espalhamento do incêndio. Um incêndio em um prédio de Madeira ($c_i = 2$), cresce e se espalha duas vezes mais rápido que um prédio de Alvenaria e Madeira ($c_i = 1$), que por sua vez também faz com que um incêndio cresça e se espalhe duas vezes mais rápido que um prédio de Alvenaria ($c_i = 0,5$) apenas.

3.1.3 Core

O módulo *Core* é responsável pela passagem de informação entre os outros módulos e entre os agentes. O *Core* controla toda a simulação, realizando chamadas ao Gerador de Incêndios e aos Agentes a cada ciclo e inclusive realizando chamadas de função na interface gráfica da simulação para exibir o estado atual da simulação.

Em cada ciclo, o *Core* executa as seguintes etapas:

1. Coleta as informações passadas por rádio pelos agentes exploradores e bombeiros.

2. Divulga para os bombeiros as informações passadas por rádio no primeiro passo. Isso faz com que as informações de rádio que os bombeiros recebem estejam sempre defasadas em um ciclo, uma vez que nos passos 3 e 4 os incêndios e prédios não estão mais no mesmo estado em que estavam quando a informação foi transmitida.

3. Executa um passo do Gerador de Incêndios, fazendo com que os incêndios existentes evoluam e se espalhem, e eventualmente criando um incêndio em local aleatório.

4. Incrementa a área queimada dos prédios, e caso toda a área do prédio seja consumida, remove o incêndio e o prédio passa a ser “queimado” e não pode mais ser alvo de incêndios.

5. Executa um passo de cada bombeiro. Em um passo, cada bombeiro pode mover-se, ou combater um incêndio ou encher o tanque em um hidrante. Caso um bombeiro esteja combatendo um incêndio, a energia do incêndio é decrementada.

6. Executa um passo de cada explorador. Em um passo, cada explorador pode apenas mover-se pelo seu setor a procura de novos incêndios.

3.1.4 Agentes

Os agentes da simulação têm como objetivo mitigar os danos causados pelos incêndios. Para isso, cada agente pode observar as células do mapa em um raio de 5 células ao redor dele, e reportar por rádio as informações obtidas. As informações de rádio chegam com um ciclo de atraso para os outros agentes, pois não seria realista se todos os agentes soubessem de tudo o que os outros sabem instantaneamente. Dentre as informações passadas por rádio estão: Ação atual do agente, os prédios avistados, e a posição e intensidade dos incêndios. A comunicação via rádio acontece através do módulo *Core*, que a cada ciclo recebe as informações baseadas no que cada agente está enxergando, e unifica todas estas informações em um objeto (removendo informações duplicadas) e o repassa para os agentes no final do ciclo. Desde o início da simulação, todos os agentes conhecem a arquitetura do mapa (localização das ruas, dos prédios e dos hidrantes), porém desconhecem os locais que possuem incêndios.

Para mover-se até o seu destino, cada agente utiliza o algoritmo A* para a escolha da rota, que conforme Russel e Norvig (2010) é simples, eficiente e o mais popular método para planejamento de rotas.

Foram definidos dois tipos de agentes: Exploradores e Bombeiros. Essa divisão é fundamental para a eficiência dos bombeiros, uma vez que estes podem focar o seu tempo no combate a incêndios, deixando a tarefa de exploração do mapa para os exploradores. A cada ciclo um agente pode realizar uma ação e reportar tudo o que observou via rádio. A única ação disponível para o agente explorador é mover-se, enquanto o bombeiro pode combater incêndio e encher o tanque de água, além da ação de movimento.

O explorador é um agente reativo, com maior velocidade de deslocamento que o bombeiro, podendo mover-se a 5 células por ciclo. Sua única tarefa é reportar tudo o que observa em cada ciclo. Cada explorador é atribuído a um setor no início da simulação e se move até ele, reportando tudo o que vê. Depois que chega ao seu setor, ele fica o

resto da simulação indo até os pontos dentro do setor que faz mais tempo que foram visitados, buscando por novos incêndios que possam ter surgido.

O bombeiro é um agente cognitivo, com inteligência baseada em RNA (Redes Neurais Artificiais). Sua principal tarefa é combater incêndios, porém, assim como o explorador, ele também reporta para os outros agentes tudo o que observa.

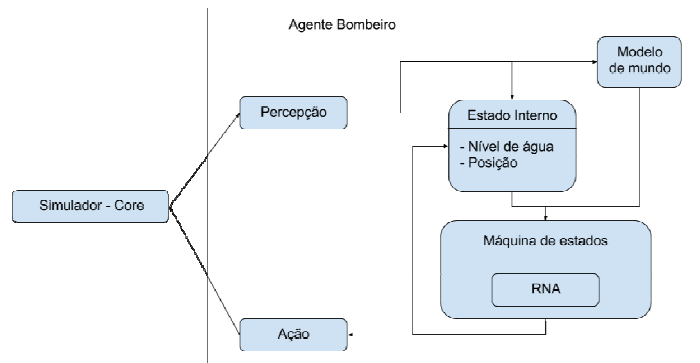


Figura 3. Arquitetura do agente bombeiro

A arquitetura proposta para o agente bombeiro pode ser observada na Figura 3. O agente recebe informações do ambiente e o resultado de suas ações através do módulo de Percepção. As percepções são utilizadas para atualizar o Modelo do mundo e a posição do agente no Estado interno. O Modelo de mundo armazena todas as informações que o agente possui sobre o ambiente, como locais com incêndios e quais incêndios os outros agentes possuem como alvo, por exemplo. O Estado interno armazena os atributos do agente. O bombeiro possui uma máquina de estados que controla o seu comportamento. As informações do Modelo do mundo e Estado interno são acessadas pela máquina de estados para determinar o próximo estado (e consequentemente, próxima ação) do agente.

Para combater incêndios, os bombeiros consomem a água que fica armazenada em seu tanque, que inicialmente está cheio. Para reabastecer o tanque de água, o bombeiro move-se até o hidrante mais próximo, onde após alguns ciclos, o tanque é completamente preenchido. A seleção de qual incêndio o bombeiro irá combater é realizada por uma RNA com aprendizagem supervisionada, descrita na próxima seção.

3.1.5 RNA

O módulo de RNA está presente nos agentes bombeiros e é responsável pela tomada de decisão do agente em relação a qual incêndio deve ser combatido por este. Para a implementação da RNA foi utilizada a biblioteca de código aberto Neuroph. Esta biblioteca permite a criação de RNAs com diferentes arquiteturas, funções de transferências, métodos de aprendizagem, métodos de cálculo de erro e critérios de parada. Para este trabalho foi criada uma rede *Multilayer Perceptron* com 1 camada oculta, utilizando como método de aprendizagem o *Backpropagation* e função de transferência sigmóide. São utilizados 7 neurônios de entrada e 1 para a saída. Cada entrada e saída é representada por uma variável normalizada do tipo *double* que pode variar de 0 até 1.

Para realizar a tomada de decisão de qual incêndio será combatido, a RNA recebe como entradas os principais fatores que determinam a importância de um incêndio: Intensidade do incêndio, material do prédio em chamas, área do prédio em chamas, número de bombeiros apagando ou indo apagar o incêndio, distância do hidrante até o incêndio, distância do bombeiro até o incêndio e média dos materiais dos vizinhos sem incêndio do prédio em chamas; e gera como saída uma nota que representa o tempo economizado e a área salva esperada no caso do agente apagar este incêndio. Estes fatores foram definidos levando em consideração os aspectos que determinam o dano que um incêndio pode causar, seja diretamente sobre o prédio que ele está, ou indiretamente através do espalhamento para prédios vizinhos. Também, são as características que auxiliam a tomada de decisão pelos bombeiros de quais focos de incêndio priorizar para combater.

Além dos 7 neurônios de entrada e 1 de saída descritos, a rede possui uma camada oculta com número de neurônios configurável pelo usuário na interface.

Antes dos agentes utilizarem a RNA para realizar tomadas de decisão ela é treinada utilizando situações exemplo onde o bombeiro pode aprender o impacto de sua seleção de qual incêndio combater, e desta maneira, aprender qual o melhor foco a ser apagado.

4. Resultados

Para aplicar a etapa de validação do sistema, foi desenvolvido um agente bombeiro mais simples, com estratégia gulosa, para comparar ao inteligente (agente com RNA). Este agente simples possui a mesma estrutura do agente inteligente, com os mesmos módulos de percepção, modelo de mundo, estado interno e ação, mas com a máquina de estados sem a RNA. Este agente simples seleciona o incêndio com menor distância até o bombeiro, sem levar em consideração a existência de setores definida pelo algoritmo de clusterização de *K-means*. As únicas situações onde os setores são considerados são no início da simulação para distribuir os bombeiros pelo mapa e quando um bombeiro não sabe da existência de incêndio para apagar, o que resulta em um comportamento de exploração dentro de seu setor.

A validação deste trabalho foi realizada com base em 10 pares de simulações, onde cada par é composto por uma simulação utilizando bombeiros com RNA e a outra utilizando bombeiros simples, ambas com os mesmos incêndios iniciais. Foram comparados os resultados de cada par de simulações e foram avaliados o total de área queimada, número de incêndios apagados, número de incêndios existentes ao final da simulação, além das informações sobre os incêndios selecionados. Cada simulação foi realizada com as mesmas configurações: 300 ciclos, em uma mapa formado por uma matriz de 100x100 células, 50 bombeiros, 10 exploradores, 10 setores, 50 incêndios iniciais, fator de evolução com valor 0,05, fator de espalhamento com valor 0,05, fator de surgimento com valor 0,05 e fator de velocidade de queima com valor 2. Os agentes com RNA utilizaram uma RNA treinada com 10.000 exemplos de treinamento, iterados por 1.000 vezes, utilizando uma taxa de aprendizagem 0,1 e 25 neurônios na camada oculta. Estes parâmetros foram definidos com base em testes que obtiveram os melhores resultados em relação à área queimada.

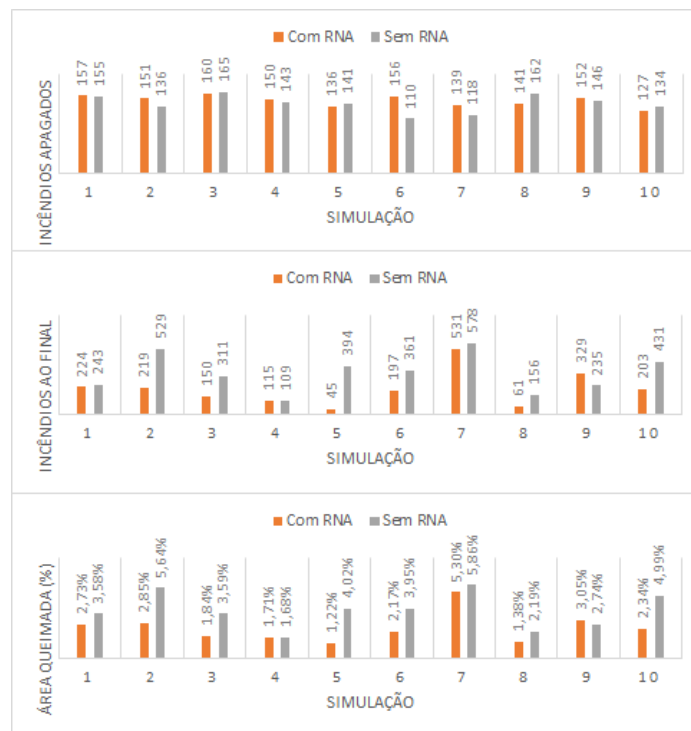


Figura 4. Gráficos de total de incêndios apagados, incêndios existentes ao final da simulação e porcentagem da área queimada ao final da simulação.

Na Figura 4 são demonstrados os gráficos com os resultados dos 10 pares de simulações. Pode-se observar que em 8 de 10 casos, os bombeiros com RNA foram mais efetivos em reduzir a área queimada pelos incêndios, e em 6 de 10 casos a quantidade de focos de incêndio combatidos pelos agentes com RNA foi maior. Apesar do total de incêndios apagados pelos agentes com e sem RNA ser semelhante em quase todas as simulações, percebe-se que o número de incêndios existentes ao final de cada simulação em geral é bem menor nos casos com RNA. Isto indica que os bombeiros inteligentes realizaram melhores escolhas de alvos para serem apagados, eliminando incêndios que se espalhariam mais facilmente.

Na Tabela 1, pode-se observar as características que a RNA considerou mais relevante na seleção do incêndio a ser combatido. Em todas as simulações a RNA selecionou em média, incêndios com menor intensidade, em prédios com menor área, material mais inflamável e com vizinhos sem incêndio mais inflamáveis (valor mais alto para o material). A preferência por incêndios com baixa intensidade acontece porque incêndios na fase inicial são os mais fáceis de serem extintos, evitando que evoluam e se espalhem causando maiores estragos. Pelo mesmo motivo incêndios em prédios menores são selecionados em média. Foi dada preferência para prédios com materiais mais inflamáveis e vizinhos mais inflamáveis porque estes são os incêndios que evoluem mais rápido e se espalham mais rapidamente, aumentando consequentemente a área queimada.

Tabela 1. Características dos incêndios selecionados em cada execução

Execução	Intensidade		Material do Prédio (Ci)		Área do Prédio		Nº de Bombeiros no incêndio		Distância Hidrante		Distância Bombeiro		Material médio dos vizinhos sem incêndios (Ci)	
	Com RNA	Sem RNA	Com RNA	Sem RNA	Com RNA	Sem RNA	Com RNA	Sem RNA	Com RNA	Sem RNA	Com RNA	Sem RNA	Com RNA	Sem RNA
1	0,269	0,459	1,662	1,104	798,024	968,991	2,607	2,798	17,478	15,149	14,087	7,078	1,292	1,103
2	0,254	0,483	1,754	1,102	883,232	1054,350	2,674	1,479	18,187	15,055	12,401	5,917	1,338	1,053
3	0,236	0,311	1,643	0,932	876,163	959,378	3,066	1,625	19,447	17,917	12,036	7,665	1,402	0,991
4	0,229	0,635	1,692	0,958	782,484	1023,749	3,747	4,672	24,470	22,464	18,454	10,451	1,338	0,956
5	0,260	0,517	1,533	1,117	861,203	966,569	4,473	2,055	25,418	16,114	20,644	7,427	1,199	1,090
6	0,300	0,606	1,504	1,085	875,641	949,311	2,990	2,125	16,227	17,405	12,137	8,009	1,239	1,053
7	0,221	0,470	1,742	0,917	910,520	1039,866	1,931	1,375	13,775	15,467	9,086	6,056	1,201	0,988
8	0,288	0,587	1,525	1,178	946,573	924,342	3,923	4,834	18,527	16,715	16,027	6,345	1,299	1,033
9	0,411	0,596	1,681	1,072	822,680	937,941	2,188	3,547	16,775	19,013	12,333	6,686	1,503	1,019
10	0,335	0,551	1,714	1,073	854,288	1063,753	2,879	1,826	17,439	20,817	16,375	7,339	1,239	1,114

5. Conclusão

O trabalho desenvolvido contribui para o trabalho de equipes de resgate durante uma situação pós-terremoto, no que diz respeito à atuação de bombeiros no combate a incêndios gerados pelo rompimento de tubulações de gás e de quedas de linhas de força. Embora o enfoque deste trabalho não tenha sido nas formas de apagar o incêndio em si, sua contribuição para este domínio é relevante, uma vez que a alocação de tarefas possui grande impacto no desempenho das equipes.

Após as análises feitas na seção de validação, concluiu-se que a estratégia aprendida pela RNA resulta em menor área queimada em relação à estratégia gulosa na maioria dos casos. O total de incêndios apagados em média pela RNA é pouco maior do que o da estratégia gulosa. Porém, apesar do número semelhante de incêndios apagados em ambas as estratégias, a RNA claramente selecionou melhor quais incêndios apagar, resultando em um número menor de incêndios existentes ao final da maioria das simulações.

Referências

- Barroso, A. D.; Santana, F. De C.; Lassance, V.; Silva, A. B. M. da; Nardin, L G.; Brandão, A. A. F.; Sichman, J. S. RoboCup Rescue 2013 LTI Agent Rescue Team Description. 2013. Disponível em: https://www.ida.liu.se/~TDDD10/papers/rsl-agents/rsl2013_submission_6.pdf.
- Basak, S.; Modanwal, N.; Mazumdar, B. D. Multi-Agent Based Disaster Management System: A Review. International Journal of Computer Science and Technology (IJCSST), Vol 2, Issue 2. 2011.
- Farinelli, A.; Grisetti, G.; Iocchi, L.; Cascio, S. L.; Nardi, D. Design and Evaluation of Multi Agent Systems for Rescue Operations. In: International Conference on Intelligent Robots and Systems, IEEE. 2003.
- Guha-Supir, D.; Below, R.; Hoyois, P.; EM-DAT: International Disaster Database. Université Catholique de Louvain – Brussels – Belgium. 2015. Disponível em: <http://www.emdat.be>.
- Jain, A. K.; Dubes, R. C. Algorithms for Clustering Data. Prentice Hall. New Jersey. 1988.
- Khouj, M.; López, C.; Sarkaria, S.; Marti, J. Disaster Management in Real Time Simulation Using Machine Learning. In: 2011 24th Canadian Conference on Electrical and Computer Engineering. IEEE. 2011. p. 1507-1510.
- Kitano, H.; Tadokoro, S. RoboCup Rescue A Grand Challenge for Multiagent and Intelligent Systems. In: AI Magazine, Vol. 22, Number 1. AAAI. 2001.

- Liang, Z.; Yang, X.; Deng, Y. A partitioning-based task allocation strategy for Police Multi-Agents. In: The 26th Chinese Control and Decision Conference. IEEE. 2014. p. 2124-2128.
- Pereira, A. H.; Nardin, L. G.; Sichman, J. S. Coordination of Agents in the RoboCup Rescue: A Partial Global approach. In: Workshop and School of Agent Systems, their Environment and Applications. IEEE. 2011. p. 45-50.
- Pessin, G.; Osório, F.; Musse, S. Simulação Virtual da Evolução de Estratégias e do Controle Inteligente em Sistemas Multi-Robóticos. In: X Symposium of Virtual and Augmented Reality. Universidade Federal da Paraíba: João Pessoa, 2008.
- Pessin, G.; Osório, F.; Musse, S.; Nonnemacher, V.; Ferreira, S. S. Utilizando Agentes Autônomos com Aprendizado para a Identificação e Combate de Incêndios em Áreas Florestais In: Anais do VII Simpósio de Informática do Planalto Médio (SIPM 2007), Universidade de Passo Fundo (UPF). 2007.
- Prado, J. P. A.; Freitas, R. L.; Guilherme, I. R.; Nakamiti, G. S.; Rillo, M. Inteligência Artificial Distribuída em Ambientes Reativos. In: I Simpósio Brasileiro de Automação Inteligente (SBAI), 1993. Disponível em: http://fei.edu.br/sbai/SBAI1993/ARTIGOS/I_SBAI_35.pdf, Junho.
- Ramchurn, S. D.; Huynh, T. D.; Ikuno, Y.; Flann, J.; We, F.; Moreau, L.; Jennings, N. R.; Fischer, J. E.; Jiang, W.; Rodden, T.; Simpson, E.; Reece, S.; Roberts, S. HAC-ER: A Disaster Response System Based on Human-Agent Collectives. In: Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015). 2015. Disponível em: <http://www.robots.ox.ac.uk/~sjrob/Pubs/fp634-ramchurn.pdf>, Junho.
- Ramchurn, S. D.; Wu, F.; Jiang, W.; Fischer, J. E.; Reece, S.; Roberts, S.; Rodden, T.; Greenhalgh, C.; Jennings, N. R. Human-agent collaboration for disaster response. In: Autonomous Agent and Multi-Agent Systems, 30. Springer. Singapura, 2016. p. 82-111.
- Ribeiro, P. C. Modelagem e Implementação OO de Sistemas Multi-Agentes. Departamento de Informática. Dissertação de Mestrado. Pontifícia Universidade Católica do Rio de Janeiro: Rio de Janeiro. 2001.
- Russel, S. J.; Norvig, P. Artificial Intelligence: A Modern Approach. 3rd Edition. Pearson, Upper Saddle River, 2010.
- Sanches, M. K. Aprendizado de máquina semi-supervisionado: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados. Dissertação de Mestrado. USP. 2003. Disponível em: http://www.teses.usp.br/teses/disponiveis/55/55134/tde-12102003-140536/publico/Dissertacao_MKS.pdf
- Singh, V. K.; Modanwal, N.; Basak, S. MAS Coordination Strategies and their Application in Disaster Management Domain. In: 2nd International Conference on Intelligent Agent and Multi-Agent Systems (IAMA), IEEE. Índia. 2011.

Applying Machine Learning to Develop an Automatic Incident Classifier: A Massive Event Case

André Accioly Lima¹, Renato Carlos Pinto¹, Leonardo Henrique Moreira^{1,2},
Célia Ghedini Ralha¹, Edison Ishikawa¹

¹Department of Computer Science – Institute of Exact Sciences
University of Brasília (UnB)
Brasília – DF – Brazil

²Systems Development Center – Brazilian Army
Brasília – DF – Brazil

{andreaccioly1, rehcarlos}@hotmail.com, moreira@cds.eb.mil.br,
{ghedini, ishikawa}@unb.br

Abstract. *Classification problems and their solving approaches constitute one of the fields of machine learning. Machine learning methods are increasingly popular to assist the decision process in incident management systems. When input data related to a great number of incidents can be generated in a short period of time, it is important to have a tool to assist the decision process in feasible time. This work applies machine learning to develop an automatic incident text classifier using FIFA Confederations Cup 2013 database. Three machine supervised learning algorithms were applied: Naive Bayes, Support Vector Machine (SVM) and Decision Trees. They were evaluated considering the ROC curve and SVM had the best performance.*

1. Introduction

The political decision to host several great events in Brazil in such short time posed major challenges for Brazilian defense and security organizations. By hosting the FIFA Confederations Cup (2013), FIFA World Cup (2014) and the Olympic Games in 2016, the defense and security Brazilian organizations were required to start the planning process before 2010. One of the main investment was the development of an information system to control and coordinate inter-agency actions. Along with it, highly reliable and available Information and Communication Technology systems were needed to provide conditions to control all activities related to security of the events.

However, if in one hand the development and employment of such tools supported an integrated environment, on the other hand the analysis of observations collected during FIFA Confederations Cup 2013 and FIFA World Cup 2014 showed an important and worrying characteristic of these great events. Under this type of operations, decision makers are exposed to information changes in intervals lower than five minutes [Moreira et al. 2016]. Under this condition, there is an urgent need for the generation of computational theories and tools to support the extraction of useful information from rapidly growing digital data [Fayyad et al. 1996].

Therefore, in order to provide enough conditions to authorities, information systems need to incorporate new features to improve situational awareness and facilitate

the decision making process. Given the magnitude of the context of massive events within the scope of public safety, the use of machine learning methods is a desirable feature of clear relevance for data, information and knowledge treatment.

In this direction, this work applies different supervised learning methods to create an automatic incident classifier for Pacificador, a tool developed by Brazilian Army to support information exchange among agents disposed in the operation area. Hence, Pacificador attends to three operational needs: situational awareness, incident treatment and actions synchronization [Moreira et al. 2016]. The experiments used the major incidents which happened during FIFA Confederations Cup 2013. All data were collected from the Pacificador database and classified in two main categories: *manifestation* and transit. Training samples were created using thesaurus techniques with characteristic terms of each category and three methods of supervised learning were evaluated: Naive Bayes[Rish 2001], Support Vector Machine (SVM)[Platt 1999] and Decision Trees [Witten et al. 2016].

The conclusion is that it is possible to create an operator support module for the Pacificador that contains an automatic long-event incident text together with statistics information. In spite of the enormous variation of texts with specific military terms and many short descriptions required by Pacificador tool, the automated classifier presents satisfactory results and create very accurate classifiers for the defined categories. Although this work focuses on two categories, this set can be extended by following the steps which were used in the design methodology.

The rest of the paper is organized as follows. A brief review of machine learning and data mining is introduced in Section 2. In Section 3, some related works are highlighted. In Section 4, the adopted methodology and the proposed solution are described. In Section 5, experiments with their evaluation are described. Finally, Section 6 highlights some final considerations and future work.

2. Background

Machine learning techniques are important for the creation of intelligent and adaptable systems, since these methods aim to generalize a solution that is adequate for the problem and has the ability to adapt to new entries. In summary, machine learning is the area of study that is concerned with how to construct computational solutions that automatically improve through their experiences [Mitchell et al. 1997].

Machine learning algorithms may follow different approaches. They depend on the application in question and the a priori knowledge that one has of the domain to which the application is part. Machine Learning systems have particular and common characteristics that allow their classification considering the language description, paradigms and forms of learning. A traditional division of approaches is: reinforcement, supervised, unsupervised and semi-supervised learning [Mitchell et al. 1997]. According to [Kulikowski and Weiss 1991], a supervised learning system is a computer program that makes decisions based on the experience contained in examples successfully solved. Thus, in this work, a supervised learning approach is applied since there was a database on incidents in Pacificador system.

Considering the various classification algorithms and their aspects of the evaluation process as presented in [Japkowicz and Shah 2011], in this work we have

applied some of the most popular supervised learning algorithms: decision trees, SVM and Naive Bayes. In the Decision Trees method, the problem is represented in a tree, where in each node the input data are analyzed and divided, performing an attribute test. The tree is traversed from top to bottom until it finds a leaf, which returns the classification for the input provided. The objective is to obtain, from a training set, the smallest possible consistent tree, so the tests that occur at each node must be organized in a way that the data is split in the best possible way [Mitchell et al. 1997].

In the SVM method there is not a priori knowledge about the application domain. This method works by creating a separator that is as far away from the training set examples as possible. The closest points of this separator are called support vectors. The SVM is a non-parametric method because it needs to save training examples for the operation of the classifier. However, in practice, you do not need to store all training data, as in other methods. What this approach does is create a linear separator between data from the different categories of the training set. There are two ways to use this method. Using the idea of *rigid margin*, a data that appears on one side of the separator is classified as that category. With a *soft margin* classifier, the distance from the separator will be analyzed, associating a greater penalty to distance, creating a larger tolerance for noisy data sets. However, not all sets will be linearly separable. In these situations, a technique called kernel trick is used. With this technique, the data is mapped into a larger dimension in which it is possible to divide them linearly. This feature makes SVM a powerful method because it is capable of representing functions of different complexities.

The third method used is Naive Bayes, which is based on probability theory. More particularly, this method operates with the idea of attribute independence. It disregards the relation that the attributes have among them, and studies the individual probability of each of them. It is a supervised learning algorithm because the training set data will be separated into different labeled classes. From the training set, the features of the document are analyzed, and then the target text is analyzed to classify the document using the naive version of the Bayes theorem. There are different possible ways to consider the features of the document being classified. In case of categorization of a document into different classes, the frequency of a particular word in document or messages is checked. Other important definitions are how the probabilities will be stored a priori, and how the words in the target text will be compared with this information obtained from the training set. The chosen form of storage can considerably affect the efficiency of the algorithm.

Another challenge that application designers generally face is how to deal with huge volume of databases and extract useful information from them. In this sense, data mining is employed to discovery patterns in data, automatically or semi-automatically, in order to create conditions to predict future situation by finding clues to already known situations. There is wide variety of tools that can be used to for data mining tasks. Regarding to open source tools, WEKA is highlighted due to its interface of easy understanding, good documentation and portability among operation systems [Witten et al. 2016].

After combining the WEKA tool with machine learning methods for processing and mining data, it is important to define metrics that can be used to evaluate performance. True positive (TP) stands for instances correctly classified as belonging to the class of interest. False positive (FP) depicts instances wrongly classified as belonging to the class

of interest. True negative (TN) deals with instances correctly classified as not belonging to the class of interest. False negative (FN) describes instances wrongly classified as not belonging to the class of interest.

These terms are usually used to compute two types of metrics. The precision value measures how many instances, among the ones classified as positive (TP and FP), do belong to the class of interest. In other words, it stands for accuracy. The recall value measures how many instances that belong to the interest class are classified as such. In other words, it stands for completeness. F-measure is a harmonic mean between these measures. Finally T_A is a measure for total accuracy [Kaur and Chhabra 2014]. All metrics are illustrated in Equation 1, 2, 3 and 4 respectively.

$$precision = \frac{TP}{TP + FP} \quad (1) \quad recall = \frac{TP}{TP + FN} \quad (2)$$

$$F = \frac{2 \times precision \times recall}{precision + recall} \quad (3) \quad T_A = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Another way to evaluate the performance of a classifier is using the Receiver Operating Characteristic (ROC) Curve. It is a plot of sensitivity versus 1-specificity. Sensitivity is the value of true positive rate (TP), also known as recall rate. Specificity is the true negative (TN) rate. What this curve exposes is the trade-off between the ability to detect cases that should be detected and false alarms. The milder the criterion of the classifier for the detection of a data as belonging to a category, the greater the chance of a false alarm.

Therefore, the performance of a classifier can be evaluated by analyzing the area under the ROC Curve since this value takes into account the different points of trade-off of the curve. The higher this value, the more the graph can achieve a high rate of true positives without increasing both the rate of false positives. In this sense, a classifier that obtained an area under equals to 0.5 would not be better than luck because it would detect the same proportion of true positives and false positives. A classifier that obtained an area equals to 1.0 would be an excellent classifier. Finally, classifiers with values below 0.5 would be worse than chance.

Furthermore, usually the text needs to be transformed into data structures that can be used efficiently by learning algorithms. It happens by transforming the textual string into a vector of words. The value contained in each vector position is an adjustment factor. *Binary presence* uses the position of the vector to indicate whether the word is present in the text fragment. In *term frequency (TF)*, each vector position corresponds to the frequency that the word appears in the fragment. In *corpus frequency*, the vector position corresponds to the frequency of the word in the total set of documents. Inverse of the frequency in the corpus, *inverse document frequency (IDF)* uses the vector position to indicate the inverse of the amount of time that the word appears in the total set of documents. It stands for the rarity of a word. It is also possible to use a combination of more than one of these measures. For instance, consider the frequency in the fragment and the inverse of the frequency in corpus - TF-IDF.

3. Related Work

There is a study with a systematic review of the applications of machine learning, data mining techniques and tools in the field of diabetes research with respect to a) Prediction and Diagnosis, b) Diabetic Complications, c) Genetic Background and Environment, and e) Health Care and Management with the first category appearing to be the most popular [Kavakiotis et al. 2017]. Authors cite that a wide range of machine learning algorithms were employed and in general 85% of the works were characterized by supervised learning approaches and 15% by unsupervised ones (more specifically, association rules). In their study the SVM arises as the most successful and widely used algorithm.

In [Kaur and Chhabra 2014] authors deal with efficient data mining procedure for predicting the diabetes from medical records of patients using a modified version of the J48 classification algorithm. The Pima Indians Diabetes Data Set was used since it collects the information of patients with and without diabetes. The data mining tool WEKA was used as an API of MATLAB for generating the J48 classifiers. The performance comparison was done including different algorithms: Naive Bayes, Multilayer Perception, Multi Class Classifier, Random Tree, REPTree, LADTree, BF Tree, AD Tree, Random Forest and J48. The experimental results showed an improvement of the proposed algorithm with 99.87% accuracy over other existing algorithms that show maximum of 77.21% accuracy. The classical J48 algorithm had 73.82% accuracy rate.

The Tweet4act system is presented in [Chowdhury et al. 2013] as a system to detect and classify crisis-(natural disasters) related messages communicated over a microblogging platform. The system extracts content features from each message and use an incident-specific dictionary to determine the period type of an incident that each message belongs to. There are three period types: pre-incident (messages talking about prevention, mitigation, and preparedness), during-incident (messages sent while the incident is taking place), and post-incident (messages related to the response, recovery, and reconstruction). Authors compared the period detection algorithm against four text-based supervised classifiers including SVM, maximum entropy (MaxEnt), Decision Trees and the ensemble classifier random forest (RF). The training model for the corpora used 80% of the data and validated the models over remaining 20% of the data by using 10-fold cross validation technique. The precision, recall and F measure of the four classifiers (SVM, MaxEnt, Tree, RF) show that the detection method can identify incident-related messages with high precision and recall, outperforming standard machine learning classification methods by a large margin of 0.69 vs. 0.44 for Joplin dataset.

Considering the related works, this article uses the three supervised classifiers algorithms: Naive Bayes, Decision Trees and SVM.

4. The Classifier Design Methodology

The overall objective of the work is to develop a module for automatic classification of incidents in major events for the Pacificador system. When a mobile agent registers an incident in such tool, it is marked on the interactive system map. This data is analyzed by the commanders of the operation, to decide how to proceed or what actions to take, helping to manage real-time incident. The module has an interface that allows automatic classification of the incident registered in a pre-established category, besides containing

statistical information about the dynamics of that type of incident. In addition, they contribute to the planning of upcoming operations.

In order to create this module, records regarding the FIFA Confederations Cup (2013) of the Pacificador database were obtained. These records contain data relating to reports and incidents that were recorded during operations that monitored game security. These data obtained have information about the date and time of the incident or report, as well as its textual description, coordinates of its location and which control center monitored it. In this way, it is possible to obtain statistics such as: in which city, incidents of a certain type have a higher occurrence, at what times and days of the week.

The use of techniques of supervised learning requires the definition of categories of the context. Originally, Pacificador has 76 possible categories, however, this division is too comprehensive and underutilized by the operators of the system. In order to reduce the number of categories, a specialist from Brazilian Army was consulted, who reduced to nine items: accreditation, crime, infrastructure, logistics, demonstration, health, terrorism, manifestation and traffic. Since there is a reduced number of incidents, a manual analysis of the data set was conducted to verify the expressivity of the categories. As a result two most expressive categories were chosen to do the experimental phase of the automatic classifier in this work: *manifestation* and *traffic*.

Therefore, the first step of the methodology applied was composed of discussion with experts, definition of categories and creation of dictionary for the categories (*thesaurus*). The unigrams of each dictionary were defined during this step with respect of their semantics and use context. The following steps were related to conceptual design.

During the second step of the methodology, the database inputs were formatted into files which contained only time, location and description of incidents. Afterwards, a text processing action was performed aiming to prepare the input data. It was necessary to remove some types of words from the original data in order to improve effectiveness of the model. Initially, a list of stopwords was defined regarding to the language in which the data was described. A stopword might be articles or pronouns which bring little information about the meaning of a sentence. In order to avoid that one word was considered different from the other just because it differed in a capital letter, all words were converted into lower case. Furthermore, there might be a number of words with the same stem and semantics but different in their suffixes. Thus, during this processing step, the text had its stopwords removed, all words were converted to lower case and only radicals remained.

After processing the input, it was performed the creation of the training set, namely, a sample of the database already classified so that the algorithm can create a generalization model. A develop program scanned the database looking for entries that had keywords belonging to a defined dictionary. The entries were separated into categories referring to this dictionary, creating a training sample.

In the third step, the WEKA mining tool was used to test the Naive Bayes, SVM and Decision Trees learning methods. Following, they were compared through measures that indicate their ability to predict new entrants, namely, precision, recall, total accuracy and ROC curves.

In the fourth step, the WEKA API was explored in other to build an user interface

which allowed the operator of Pacificador to immediately obtain the classification for the new inserted incident. From the category detection, relevant statistics on the incident dynamics were also retrieved.

5. Evaluation

After using a training set to obtain a classifier model, it was necessary to proceed to evaluation phase. The test environment used to validate the proposal included a CPU AMD E-450 @ 1.65 GHz, 2GB RAM with Ubuntu 13.04. The WEKA tool used was version 3.8.0 and Snowball Stemmer English was applied for text processing. Inputs was based on FIFA Confederations Cup 2013 Pacificador database, totaling 811 instances.

Two experiments were performed. First, the inputs were checked about their classification in the *manifestation* category. Following, from the instances that were not considered part of this class, a second evaluation was done considering the *traffic* category.

In order to evaluate Naive Bayes supervised learning methods, SVM and Decision Trees, their F-weighted measures were evaluated. The values in F-measure recorded ranges from 0 to 1, hence, a value of 0.91 represents a percentage of 91%. This measure only reached a high value if the precision and recall value are high. The methods tested were Naive Bayes [Rish 2001], Decision Trees (J48) as a C4.5 algorithm implementation available at WEKA[Witten et al. 2016] and SVM with SMO optimization [Platt 1999]. For each method the weights TF, IDF, TF-IDF and Binary were applied and the minimum frequency (minimum amount of occurrence of terms in the set total to be considered) of the words was tested with values of 1, 3 and 5. The values of minimum frequency 2 and 4 were not tested because it was preferred to adopt a greater interval between these values. Values greater than 5 were also not tested because they would restrict vocabulary too much.

The cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model and a test set to evaluate it. The ten-fold cross-validation break data into 10 sets of size $n/10$, train on 9 datasets and test on 1, and repeat 10 times and take a mean accuracy. [Kohavi et al. 1995] reported on a largescale experiment over half a million runs of C4.5 and a Naive-Bayes algorithm to estimate the effects of different parameters on these algorithms on real-world datasets and state that the ten-fold cross-validation may be better than the more expensive leaveone-out cross-validation. Thus we have used ten-fold cross-validation.

The experiments with *manifestation* and *traffic* categories are reported in the sequence.

5.1. Manifestation category

The training set of the *manifestation* category was obtained through the dictionary of items, such as *manifest*, *marcha*, *passeata*, *protest*, *turb*, (and variations, e.g, *Manifest*, *MANIFEST*), obtaining a total of 187 files for this category, leaving 624 files for the *no manifestation* category. The total of these sample files had 1622 different terms¹.

Table 1 presents the result, showing the values of F-measure of the three methods with weights TF, IDF, TF-IDF and Binary for this category. Regarding to Naive Bayes

¹Items and terms are used as synonymous

method, it is possible to notice that the binary weighting had the best performance among the weights tested for all variations of minimum frequency, and had its highest value of F-measure for the minimum frequencies 3 and 5 reaching a value of 0.91. However, for the IDF weight, F-measure reached its highest value with minimum requirement 1, and for TF and TF-IDF, with a minimum frequency of 3. The recorded results were good, regardless of vocabulary constraint (minimum frequency variation) and measurement values were above 90%, indicating that a classifier with these parameters had a good predictive capacity for the category of a new incident.

The result from Decision Trees method (0.958) regardless weighting or minimum frequency was higher than the one with the best parameter configuration of Naive Bayes method, which achieved for the minimum frequencies of 3 and 5 and binary weighting the result of 0.91.

Concerning the SVM method, all values were higher than those from Naive Bayes, except for the configuration of minimum frequency 1 and IDF weighting that resulted in 0.952. All other configurations achieved values higher than those from Decision Trees.

Table 1. F-measure for manifestation.

	Naive Bayes			DT	SVM		
	Minimum frequency						
	1	3	5	1, 3, 5	1	3	5
TF	0.907	0.908	0.905	0.958	0.983	0.990	0.995
IDF	0.907	0.903	0.904	0.958	0.952	0.978	0.988
TF-IDF	0.907	0.908	0.905	0.958	0.983	0.990	0.995
Binary	0.909	0.910	0.910	0.958	0.989	0.989	0.995

5.2. Traffic category

From the 624 files of *no manifestation* category, we set out to define the second category. Similarly, the *traffic* category was chosen after a manual analysis of the data. The sample obtained represents 12% of the set of 624 files, and presents characteristic terms that facilitate the detection of an incident as being of the category. The training set of the *traffic* category was obtained through the dictionary of items (total of 51 terms) and a total of 81 files were obtained for this category, with 543 files remaining for the *no traffic* category. The total amount of these files was 1233 different terms.

Table 2 shows the values of F-measure of the three methods with weights TF, IDF, TF-IDF and Binary for the *traffic* category. Regarding to Naive Bayes method, it is possible to notice that the values of F-measure obtained with the Binary weighting were better (0.918, 0.919 and 0.922) than the other weights in all the minimum frequency variations, and they enhanced as the increase of the minimum frequency of terms, that is, the values grew as the vocabulary becomes more restricted. It is interesting to note that the values of F-measures of the TF and TF-IDF weights coincided in all variations of the minimum frequency. The results recorded were good because, regardless of vocabulary restriction (minimum frequency variation) and weighting, the values of F-measure were above 90%, indicating that a classifier with these parameters has a good prediction capacity of the category of a new incident.

Concerning the Decision Trees J48 method, similarly to the results of the *manifestation* category, part of the results obtained were equal, regardless of the minimum

frequency variation of terms. For the minimum frequencies of terms 1 and 3, regardless of weighting variation, the result was 0.957 and for the minimum frequency 5, the result was 0.961. Therefore, restricting vocabulary increased the value of F-measure to the Decision Trees method. It can also be noticed that the lowest value obtained (0.957) is higher than the best value obtained with the Naive Bayes method (0.922).

Lastly, the SVM method had the values of F-measure greater than the methods Naive Bayes and Decision Trees J48. The lowest value obtained from 0.966 was greater than the best Decision Trees value that was 0.961. Different from the SVM experiment for the *manifestation* category that had the best results with the minimum frequency 5, in this experiment the minimum frequency value 3 was the one with better F-measure results. In particular, with a minimum frequency of 3 and using the TF-IDF weight, the highest value of this measure reached 0.987. This high value for F-measure meant that the classifier with these parameters had a high value of both precision and recall, indicating an excellent ability to predict the category of a new incident.

Table 2. F-measure for traffic.

	Naive Bayes			DT			SVM		
	Minimum frequency								
	1	3	5	1	3	5	1	3	5
TF	0.902	0.904	0.905	0.957	0.957	0.961	0.975	0.985	0.966
IDF	0.909	0.908	0.913	0.957	0.957	0.961	0.977	0.984	0.966
TF-IDF	0.902	0.904	0.905	0.957	0.957	0.961	0.975	0.987	0.966
Binary	0.918	0.919	0.922	0.957	0.957	0.961	0.975	0.982	0.968

5.3. Results analysis

After obtaining the data related to the results of each machine learning method for each category, it was possible to highlight the best algorithm for each case from the perspective of F-measure.

Table 3 outlines the comparative results of the *manifestation* and *traffic* categories. In both cases, it was possible to notice that the SVM method had the best results, while the Naive Bayes method presented the worst F-measures.

Figure 1 compares the performance of the classifiers with the worst and the best F-measure. Regarding to the *manifestation* category, Figure 1(A) shows the ROC curve for a Naive Bayes classifier with IDF weighting and minimum frequency 3 whereas Figure 1(B) outlines curve for SVM method with TF-IDF weighting and minimum frequency 5. Concerning the *traffic* category, Figure 1(C) shows the ROC curve for a Naive Bayes

Table 3. F-measure and Total Accuracy (T_A) for both categories.

	<i>Manifestation</i>		<i>Traffic</i>	
	F	T_A	F	T_A
Naive Bayes	0.910	0.9074	0.923	0.9199
J48	0.955	0.9593	0.961	0.9632
SVM	0.995	0.9951	0.987	0.9872

classifier with TF weighting and minimum frequency 1 whereas Figure 1(D) outlines curve for SVM method with TF-IDF weighting and minimum frequency 3. It is interesting to note that even the classifiers with the lowest F-measures still had high values of this measure, hence a good performance considering the perspective of the area under the ROC curve. The best-fit measuring device F, in Figure 1(B), achieved excellent performance from the perspective of the area under the ROC Curve. A high value of the Measure Area under the ROC Curve indicates that the classifier has a high value of true positives, with a low false positive rate.

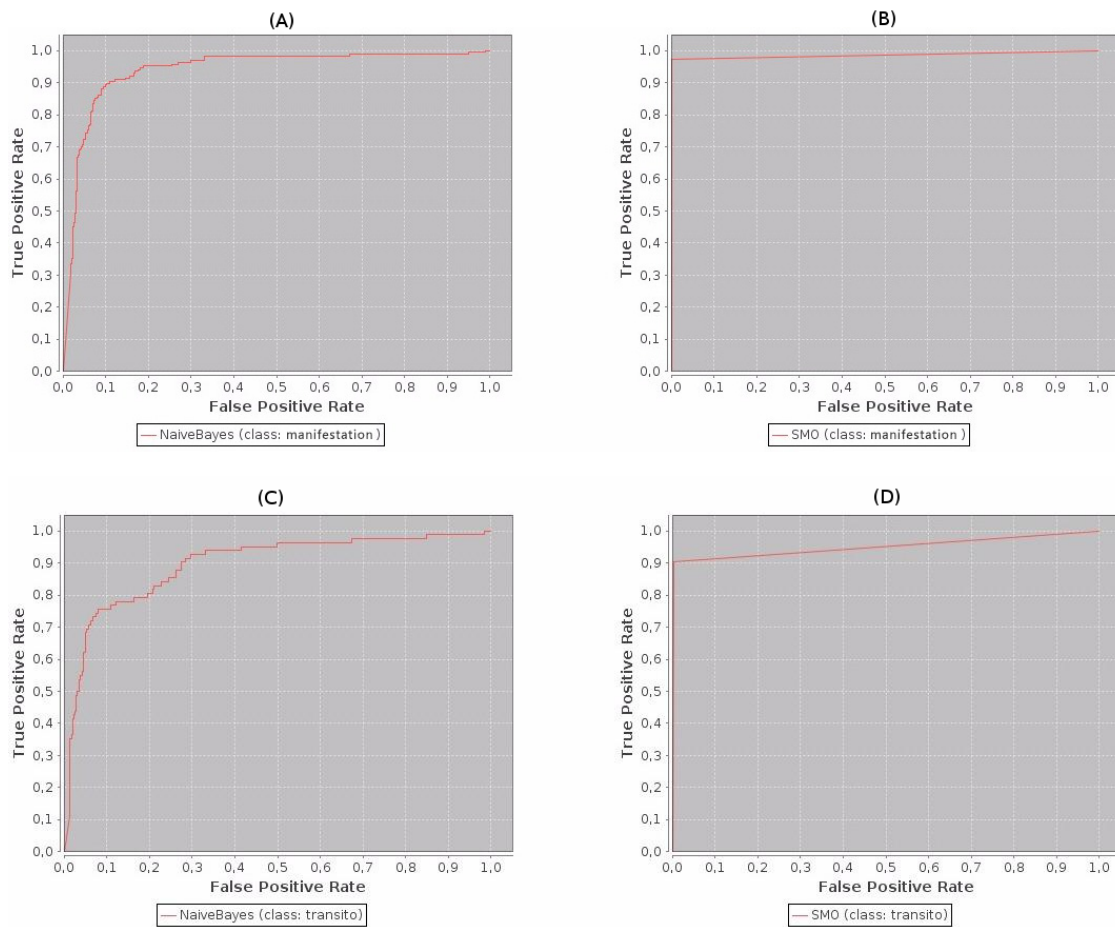


Figure 1. Comparison of ROC Curves for each category (A): Manifestation - ROC Curve for Lower F-measure Setup. (B): Manifestation - ROC Curve for Greater F-measure Setup. (C): Traffic - ROC Curve for Lower F-measure Setup. (D): Traffic - ROC Curve for Greater F-measure Setup.

In general, the results obtained with the *manifestation* category were better than the results of the *traffic* category. This is due to the fact that the first category had a larger database for learning than the second. The *manifestation* category accounted for 23% of its total (811 files), while the *traffic* category accounted for 12% of its total (624 files). In addition, the variation within the database of the *traffic* category itself was larger with a wider dictionary with 17 characteristic terms, whereas the dictionary of the *manifestation* category had only 5 terms. This variation impaired the creation of a prediction model.

However, for both categories, the F-measures presented had good values by the

fact of having very characteristic terms, which helped the class detection performed by the algorithms.

It is also interesting to note that for both categories, the SVM method was the best measure method, however, varying the configuration parameters. For the *manifestation* category, the SVM method reached its highest value with the minimum frequency 5 and weights TF, TF-IDF or Binary. In the *transit* category, the SVM reached the best value with minimum frequency 3 and TF-IDF weighting.

6. Conclusion

This work developed an automatic classifier solution for incident texts. The WEKA tool was used to study the application of learning methods in incident texts of the FIFA Confederations Cup 2013, obtained from the Pacificador system database. In the experiments carried out, two categories were defined namely, *manifestation* and *traffic*, training samples were created from dictionaries with characteristic terms of each category and methods of supervised learning were compared to evaluate the possibility of creating an automatic classifier of text of incidents. Three methods were evaluated: Naive Bayes, SVM and Decision Trees.

For the *manifestation* category, the best configurations were those of TF-IDF, TF and Binary weighting with minimum frequency of occurrence 5 with the SVM method. For the *traffic* category, the best configuration was with TF-IDF weighting, minimum frequency of occurrence of terms 3 also with the SVM method. For these better configurations, the ROC curve generated was also analyzed, confirming an optimum performance.

With the analysis of the best configuration for each category, an interface based on TF-IDF was created to provide the classification of new incident. Therefore, it is possible to create a support module that contains an automatic long-event incident text sorter with past event statistics. In spite of the enormous variation of texts with specific military terms and many short descriptions, the methods employed, with tools like stemmers and removal of stopwords, can obtain a very satisfactory result and create accurate classifiers for the defined categories.

Regarding to future works, definitions of standard operational procedure and continuous update of statistics for each category can be highlighted in order to assist authorities during decision making. Furthermore, new categories can be defined to improve the representation of incidents. It is also desirable the evaluation of algorithm combination, allowing the detection of anomalous events, and experimentation with other massive events.

References

- [Chowdhury et al. 2013] Chowdhury, S. R., Imran, M., Asghar, M. R., Amer-Yahia, S., and Castillo, C. (2013). Tweet4act: Using incident-specific profiles for classifying crisis-related messages. In *10th International ISCRAM Conference*.
- [Fayyad et al. 1996] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37.

- [Japkowicz and Shah 2011] Japkowicz, N. and Shah, M. (2011). *Evaluating learning algorithms: a classification perspective*. Cambridge University Press.
- [Kaur and Chhabra 2014] Kaur, G. and Chhabra, A. (2014). Improved j48 classification algorithm for the prediction of diabetes. *International Journal of Computer Applications*, 98(22).
- [Kavakiotis et al. 2017] Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., and Chouvarda, I. (2017). Machine learning and data mining methods in diabetes research. *Computational and Structural Biotechnology Journal*.
- [Kohavi et al. 1995] Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial intelligence IJCAI*, volume 2, pages 1137–1145. Stanford, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc.
- [Kulikowski and Weiss 1991] Kulikowski, C. A. and Weiss, S. M. (1991). Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems. *Machine Learning*. San Francisco: Morgan Kaufmann Publishers. ISBN, pages 1–55860.
- [Mitchell et al. 1997] Mitchell, T. M. et al. (1997). Machine learning. WCB.
- [Moreira et al. 2016] Moreira, L. H., Camilo, M. J., Kohl, A., Rocha, C. E. P., and Ralha, C. G. (2016). Performance evaluation of c2 centers and model for information interarrival process. In *Proc. 21st International Command and Control Research and Technology Symposium (ICCRTS)*.
- [Platt 1999] Platt, J. C. (1999). 12 fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods*, pages 185–208.
- [Rish 2001] Rish, I. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York.
- [Witten et al. 2016] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Simple Interactive Image Segmentation using Label Propagation through kNN graphs

Fabricio A. Breve¹

¹Instituto de Geociências e Ciências Exatas
Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP)
Avenida 24A, 1515 – 13.506-900 – Rio Claro – SP – Brazil

fabricio@rc.unesp.br

Abstract. *Many interactive image segmentation techniques are based on semi-supervised learning. The user may label some pixels from each object and the SSL algorithm will propagate the labels from the labeled to the unlabeled pixels, finding object boundaries. This paper proposes a new SSL graph-based interactive image segmentation approach, using undirected and unweighted kNN graphs, from which the unlabeled nodes receive contributions from other nodes (either labeled or unlabeled). It is simpler than many other techniques, but it still achieves significant classification accuracy in the image segmentation task. Computer simulations are performed using some real-world images, extracted from the Microsoft GrabCut dataset. The segmentation results show the effectiveness of the proposed approach.*

1. Introduction

Image segmentation is considered one of the most difficult tasks in image processing [Gonzalez and Woods 2008]. It is the process of dividing an image into parts, identifying objects or other relevant information [Shapiro and Stockman 2001]. Fully automatic segmentation is still very difficult to accomplish and the existing techniques are usually domain-dependent. Therefore, interactive image segmentation, in which the segmentation process is partially supervised, has experienced increasing interest in the last decades [Boykov and Jolly 2001, Grady 2006, Protiere and Sapiro 2007, Blake et al. 2004, Ducournau and Bretto 2014, Ding and Yilmaz 2010, Rother et al. 2004, Paiva and Tasdizen 2010, Li et al. 2010, Artan and Yetik 2010, Artan 2011, Xu et al. 2008, Breve et al. 2015b, Breve et al. 2015a].

Semi-supervised learning (SSL) is an important field in machine learning, usually applied when unlabeled data is abundant but the process of labeling is expensive, time consuming and/or requiring intensive work of human specialists [Zhu 2005, Chappelle et al. 2006]. This characteristics makes SSL an interesting approach to perform interactive image segmentation, which may be seen as a pixel classification process. In this scenario, there are often many unlabeled pixels to be classified. An human specialist can easily classify some of them, which are away from the borders, but the process of defining the borders manually is difficult and time consuming.

Many interactive image segmentation techniques are, in fact, based on semi-supervised learning. The user may label some pixels from each object, away from the boundaries where the task is easier. Then, the SSL algorithm will iteratively propagate the labels from the labeled pixels to the unlabeled pixels, finding the boundaries. This

paper proposes a different SSL-based interactive image segmentation approach. It is simpler than many other techniques, but it still achieves significant classification accuracy in the image segmentation task. In particular, it was applied to some real-world images, including some images extracted from the Microsoft GrabCut dataset [Rother et al. 2004]. The segmentation results show the effectiveness of the proposed approach.

1.1. Related work

The approach proposed in this paper may be classified in the category of graph-based semi-supervised learning. Algorithms on this category rely on the idea of building a graph which nodes are data items (both labeled and unlabeled) and the edges represent similarities between them. Label information from the labeled nodes is propagate through the graph to classify all the nodes [Chapelle et al. 2006]. Many graph-based methods [Blum and Chawla 2001, Zhu et al. 2003, Zhou et al. 2004, Belkin et al. 2004, Belkin et al. 2005, Joachims 2003] are similar and share the same regularization framework [Zhu 2005]. They usually employ weighted graphs and labels are spread globally, differently from the proposed approach, where the label spreading is limited to neighboring nodes and the graph is undirected and unweighted.

Another graph-based method, known as Label Propagation through Linear Neighborhoods [Wang and Zhang 2008], also uses a k -nearest neighbors graph to propagate labels. However, the edges have weights, which require the resolution of quadratic programming problems to be calculated, prior to the iterative label propagation process. On the other hand, the proposed approach uses only unweighted edges.

1.2. Technique overview

In the proposed method, an unweighted and undirected graph is generated by connecting each node (data item) to its k -nearest neighbors. Then, in a iterative process, unlabeled nodes will receive contributions from all its neighbors (either labeled or unlabeled) to define their own label. The algorithm usually converges quickly, and each unlabeled node is labeled after the class from which it received most contributions. Differently from many other graph-based methods, no calculation of edge weights or Laplacian matrix are required.

2. The Proposed Model

In this section, the proposed technique will be detailed. Given a bidimensional digital image, the set of pixels are reorganized as $\mathfrak{X} = \{x_1, x_2, \dots, x_L, x_{L+1}, \dots, x_N\}$, such that $\mathfrak{X}_L = \{x_i\}_{i=1}^L$ is the labeled pixel subset and $\mathfrak{X}_U = \{x_i\}_{i=L+1}^N$ is the unlabeled pixels subset. $\mathfrak{L} = \{1, \dots, C\}$ is the set containing the labels. $y : \mathfrak{X} \rightarrow \mathfrak{L}$ is the function associating each $x_i \in \chi$ to its label $y(x_i)$ as the algorithm output. The algorithm will estimate $y(x_i)$ for each unlabeled pixel $x_i \in \mathfrak{X}_U$.

2.1. k -NN Graph Generation

A large amount of features may be extracted from each pixel x_i to build the graph. In this paper, 23 features are used. They are shown on Table 1. These are the same features used in [Breve 2015].

For measures 12 to 23, the pixel neighbors are the 8-connected neighborhood, except on the borders where no wraparound is applied. All components are normalized

Table 1: List of features extracted from each image to be segmented

#	Feature Description
1	Pixel row location
2	Pixel column location
3	Red (R) component of the pixel
4	Green (G) component of the pixel
5	Blue (B) component of the pixel
6	Hue (H) component of the pixel
7	Saturation (S) component of the pixel
8	Value (V) component of the pixel
9	ExR component of the pixel
10	ExG component of the pixel
11	ExB component of the pixel
12	Average of R on the pixel and its neighbors (MR)
13	Average of G on the pixel and its neighbors (MG)
14	Average of B on the pixel and its neighbors (MB)
15	Standard deviation of R on the pixel and its neighbors (SDR)
16	Standard deviation of G on the pixel and its neighbors (SDG)
17	Standard deviation of B on the pixel and its neighbors (SDB)
18	Average of H on the pixel and its neighbors (MH)
19	Average of S on the pixel and its neighbors (MS)
20	Average of V on the pixel and its neighbors (MV)
21	Standard deviation of H on the pixel and its neighbors (SDH)
22	Standard deviation of S on the pixel and its neighbors (SDS)
23	Standard deviation of V on the pixel and its neighbors (SDV)

to have mean 0 and standard deviation 1. They are also scaled by a vector of weights λ in order to emphasize/deemphasize each feature during the graph generation. ExR, ExG, and ExB components are obtained from the RGB components using the method described in [Lichman 2013]. The HSV components are obtained from the RGB components using the method described in [Smith 1978].

The undirected and unweighted graph is defined as $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V} = \{v_1, v_2, \dots, v_N\}$ is the set of nodes, and \mathbf{E} is the set of edges (v_i, v_j) . Each node v_i corresponds to a pixel x_i . Two nodes v_i and v_j are connected if v_j is among the k -nearest neighbors of v_i , or vice-versa, considering the Euclidean distance between x_i and x_j features. Otherwise, v_i and v_j are disconnected.

2.2. Label Propagation

For each node v_i , a domination vector $\mathbf{v}_i^\omega(\mathbf{t}) = \{v_i^{\omega_1}(t), v_i^{\omega_2}(t), \dots, v_i^{\omega_C}(t)\}$ is created. Each element $v_i^{\omega_c}(t) \in [0, 1]$ corresponds to the domination level from the class c over the node v_i . The sum of the domination vector in each node is always constant, $\sum_{c=1}^C v_i^{\omega_c} = 1$.

The domination levels are constant in nodes corresponding to labeled pixels, with full domination by the corresponding class. On the other hand, domination levels are variable in nodes corresponding to unlabeled pixels and they are initially set equally among

classes. Therefore, for each node v_i , the domination vector \mathbf{v}_i^ω is set as follows:

$$v_i^{\omega_c}(0) = \begin{cases} 1 & \text{if } x_i \text{ is labeled and } y(x_i) = c \\ 0 & \text{if } x_i \text{ is labeled and } y(x_i) \neq c \\ \frac{1}{C} & \text{if } x_i \text{ is unlabeled} \end{cases} . \quad (1)$$

In the iterative phase, at each iteration each unlabeled node will get contributions from all its neighbors to calculate its new domination levels. Thus, for each unlabeled node v_i , the domination levels are updated as follows:

$$\mathbf{v}_i^\omega(\mathbf{t} + 1) = \frac{1}{K} \sum_{j \in N(v_i)} \mathbf{v}_j^\omega(\mathbf{t}), \quad (2)$$

where $K = |N(v_i)|$ is the size of $N(v_i)$, and $N(v_i)$ is the set of the v_i neighbors. In this way, the new dominance vector \mathbf{v}_i^ω is the arithmetic mean of all its neighbors dominance vectors, no matter if they are labeled or unlabeled.

The average maximum domination levels is defined as follows:

$$\langle v_i^{\omega_m} \rangle, m = \arg \max_c v_i^{\omega_c}, \quad (3)$$

considering all v_i representing unlabeled nodes. $\langle v_i^{\omega_m} \rangle$ is checked every 10 iterations and the algorithm stops when its increase is below 0.001 between checkpoints.

At the end of the iterative process, each unlabeled pixel is assigned to the class that has the highest domination level on it:

$$y(x_i) = \arg \max_c v_i^{\omega_c} \quad (4)$$

2.3. The Algorithm

Overall, the proposed algorithm can be outlined as follows:

Algorithm 1: The proposed method algorithm

- 1 Build the k -NN graph, as described in Subsection 2.1;
 - 2 Set nodes' domination levels by using Eq. (1);
 - 3 **repeat**
 - 4 **for each unlabeled node do**
 - 5 Update node domination levels by using Eq. (2);
 - 6 **until the stopping criterion is satisfied;**
 - 7 Label each unlabeled pixel using Eq. (4);
-

3. Implementation

In order to reduce the computational resources required by the proposed method, the following implementation strategy is applied.

The iterative step of the algorithm is very fast in comparison with the graph generation step, i.e., the graph generation dominates the execution time. Therefore, the graph

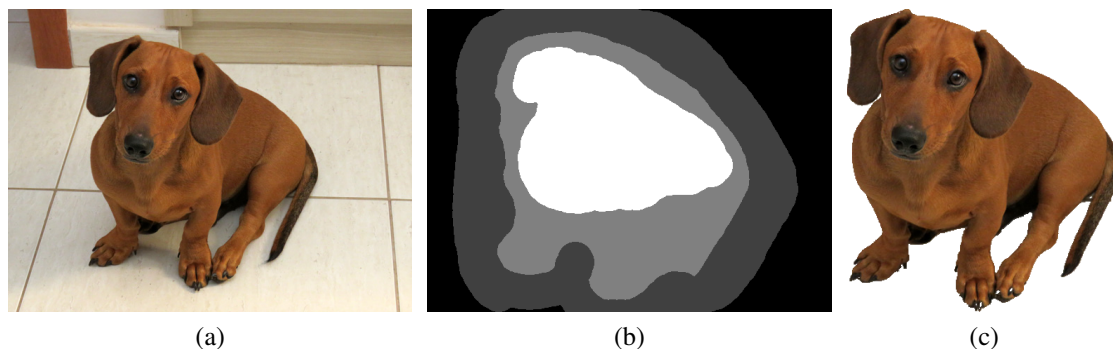


Figure 1: (a) Original image, (b) Trimap providing seed regions for Fig. 1a segmentation, (c) Close-up foreground segmentation results by the proposed method.

is generated using the k -d trees method [Friedman et al. 1977], so the algorithm runs in linearithmic time ($O(N \log N)$).

In the iterative step, each iteration runs in $O(uk)$, where u is the amount of unlabeled nodes and k is usually proportional to the amount of neighbors each node has (not equal because the graph is undirected). u is usually a fraction of N in practical problems, and often $k \ll n$. By increasing k , one also increases each iteration execution time. On the other hand, the amount of iterations required to converge decreases as the graph becomes more connected and the labels propagate faster, as it was empirically observed in computer simulations.

The iterative steps are synchronous, i.e., the contributions any node receives to produce its domination vector in time $t + 1$ refer to the domination levels its neighbors had in time t . Therefore, parallelization of this step, corresponding to the inner loop in steps 4 and 5 of the Algorithm 1, is possible. Nodes can calculate their new domination vectors in parallel without running into race conditions. Synchronization is only required between iterations of the outer loop (steps 3 to 6).

4. Experiments

The proposed technique efficacy is first tested using the real-world image shown on Fig. 1a, extracted from [Breve et al. 2015b], which has 576×432 pixels. A trimap providing seed regions is presented in Figure 1b. Black (0) represents the background, ignored by the algorithm; dark gray (64) is the labeled background; light gray (128) is the unlabeled region, which labels will be estimated by the proposed method; and white (255) is the labeled foreground.

The proposed technique efficacy is then verified using a series of computational experiments using nine image selected from the Microsoft GrabCut database [Rother et al. 2004]¹. The selected images are shown on Fig. 2. The corresponding *trimaps* providing seed regions are shown on Fig. 3. Finally, the *ground truth* images are shown on Fig. 4.

¹Available at <http://web.archive.org/web/20161203110733/research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm>



Figure 2: Original images from the GrabCut dataset: (a) 21077; (b) 124084; (c) 271008; (d) 208001; (e) llama; (f) doll; (g) person7; (h) sheep; (i) teddy.

For each image, k and the vector of weights λ were optimized using the genetic algorithm available in Global Optimization Toolbox of MATLAB, with its default parameters.

5. Results and Discussion

First, the proposed method was applied to the image shown on Fig. 1a. The best segmentation result is shown on Fig. 1c. By comparing this output with the segmentation result achieved in [Breve et al. 2015b] for the same image, one can notice that the proposed method achieved slightly better results, by eliminating some misclassified pixels and better defining the borders.

Then, the proposed method was applied to the nine images shown on Fig. 2, as described on Section 4. The best segmentation results achieved with the proposed method are shown on Fig. 5. Error rates are computed as the fraction between the amount of incorrectly classified pixels and the total amount of unlabeled pixels (light gray on the *trimaps* images shown on Fig. 3). Notice that *ground truth* images (Fig. 4) have a thin contour of gray pixels, which corresponds to uncertainty, i.e., pixels that received different

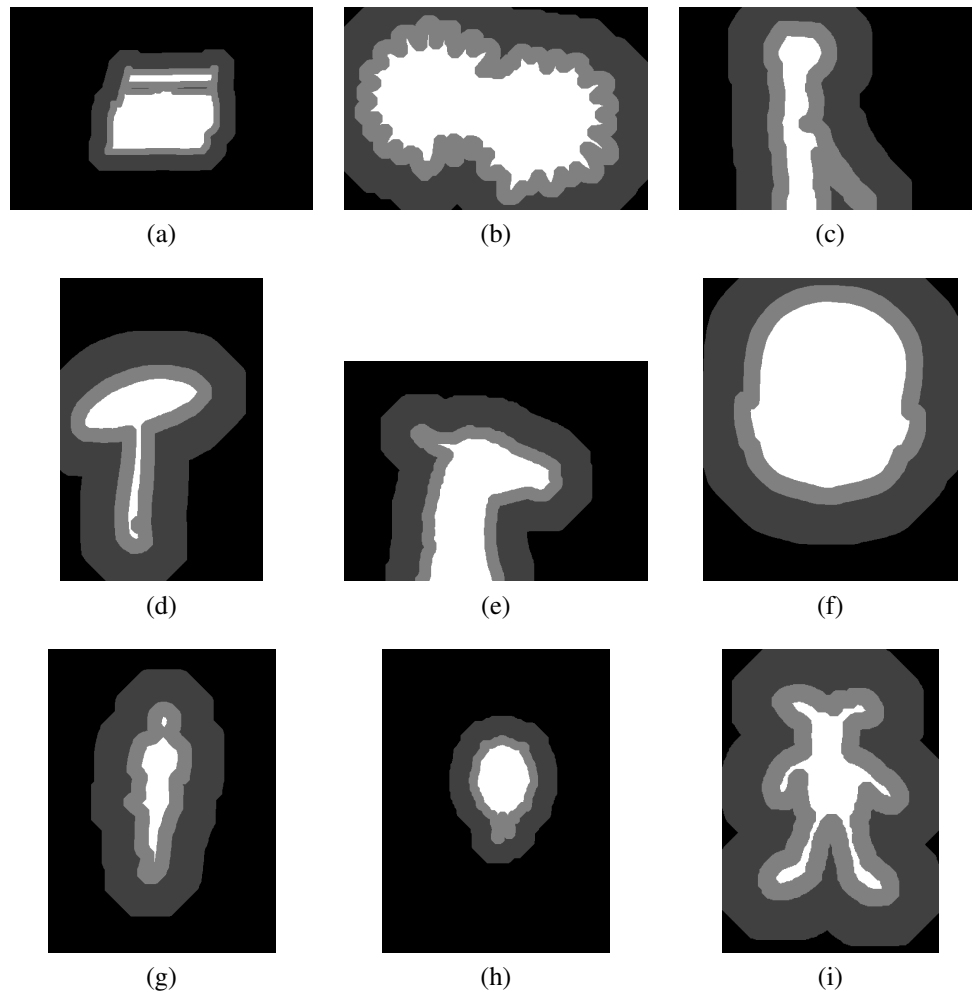


Figure 3: The trimaps providing seed regions from the GrabCut dataset: (a) 21077; (b) 124084; (c) 271008; (d) 208001; (e) llama; (f) doll; (g) person7; (h) sheep; (i) teddy.

labels by the different persons who did the manual classification. These pixels are not used in the classification error calculation.

Segmentation error rates are also summarized on Table 2. Some results from other methods [Ducournau and Bretto 2014, Ding and Yilmaz 2008, Breve et al. 2015b] are also included for reference. By analyzing them, one can notice that the proposed method has comparable results. The results from the other methods were extracted from the respective references.

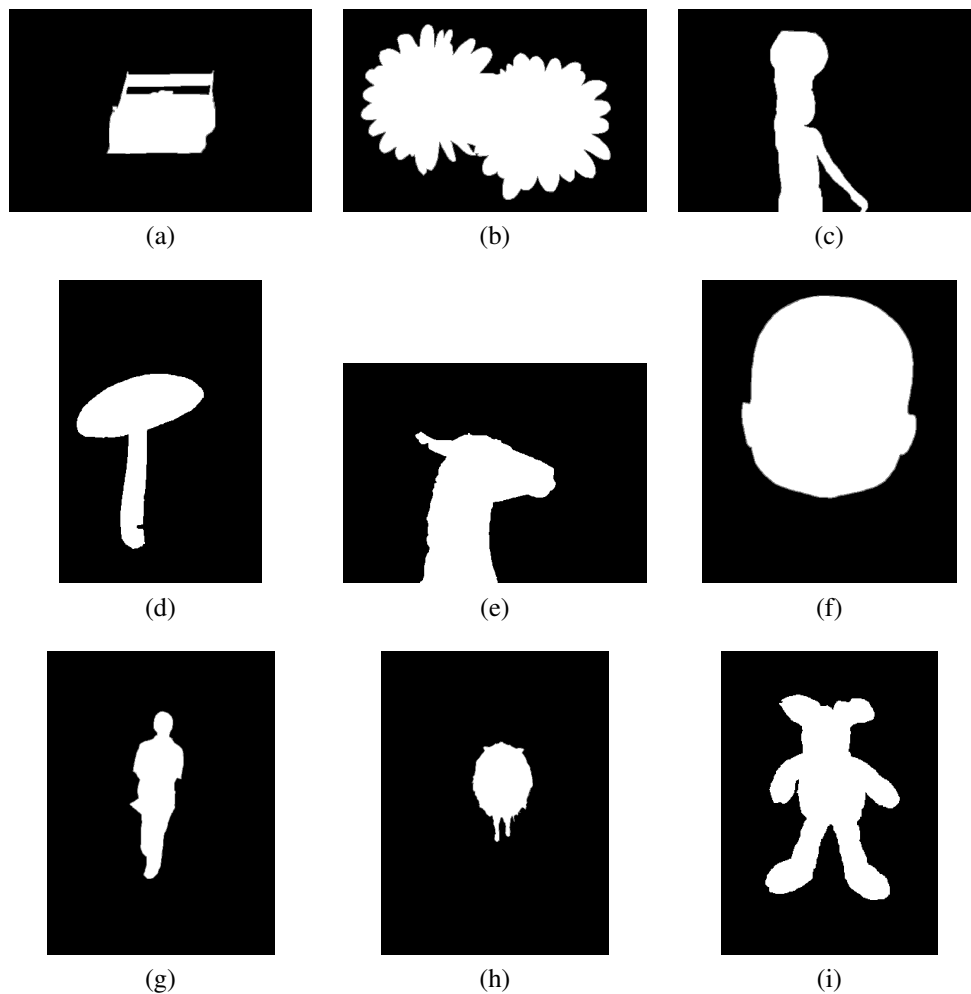


Figure 4: Close-up foreground segmentation results by the proposed method: (a) 21077; (b) 124084; (c) 271008; (d) 208001; (e) llama; (f) doll; (g) person7; (h) sheep; (i) teddy.

Table 2: Segmentation error rates achieved by Learning on Hypergraphs model (ISLH) [Ding and Yilmaz 2008], Directed Image Neighborhood Hypergraph model (DINH) [Ducournau and Bretto 2014], Particle Competition and Cooperation (PCC) [Breve et al. 2015b] and the proposed model (LPKNN).

Image	ISLH	DINH	PCC	LPKNN
21077 (cars)	13.24%	-	-	5.20%
124084 (flowers)	-	3.51%	1.09%	0.57%
271008 (children)	-	5.12%	7.67%	3.09%
208001 (mushroom)	-	3.65%	6.19%	3.88%
llama	-	-	-	6.72%
doll	-	-	-	0.06%
person7	-	2.82%	1.53%	0.95%
sheep	-	5.85%	1.24%	0.67%
teddy	4.56%	-	-	1.48%

It is also important to notice that the proposed method is deterministic. Given the same parameters, it will always output the same segmentation result on different execu-



Figure 5: The ground-truth images from the GrabCut dataset: (a) 21077; (b) 124084; (c) 271008; (d) 208001; (e) llama; (f) doll; (g) person7; (h) sheep; (i) teddy. Error rates are indicated below each image.

tions. Other methods, like Particle Competition and Cooperation [Breve et al. 2015b], are stochastic. Therefore, they may output different segmentation results on each execution.

The optimized parameters k and features weights (λ) are shown on Table 3. Considering the 10 images evaluated in this paper, pixel location features (Row and Col) are the most important features, followed by the ExB component, intensity (V), and the mean of green (MG). The least important features were hue (H), saturation (S) and all those related to standard deviation. However, no single feature received a high weight in all images. The optimal weights and k seem to be highly dependent on image characteristics.

Table 3: Parameter k and feature weights λ optimized by the proposed method for each segmented image.

Image / Feature	dog	21077 (cars)	124084 (flowers)	271008 (children)	208001 (mushroom)	llama	doll	person7	sheep	teddy	Mean	Standard Deviation
K	80	1746	32	24	3	29	409	175	165	16	267.90	533.86
Row	0.9907	0.9624	0.9916	0.4440	0.9778	0.2270	0.9733	0.7895	0.9441	0.4417	0.7742	0.2904
Col	0.8037	0.7163	0.6668	0.9142	0.9699	0.9955	0.3366	0.9858	0.8308	0.3941	0.7614	0.2368
R	0.6329	0.0348	0.0550	0.9109	0.4452	0.4467	0.5563	0.5846	0.8823	0.2783	0.4827	0.2999
G	0.3503	0.4935	0.2452	0.9633	0.2972	0.2759	0.3098	0.1000	0.5431	0.6671	0.4246	0.2500
B	0.9568	0.4486	0.1784	0.3610	0.5355	0.4602	0.2596	0.1883	0.7959	0.3919	0.4576	0.2524
H	0.0252	0.0445	0.0782	0.3739	0.0662	0.7357	0.0253	0.1472	0.1062	0.1409	0.1743	0.2220
S	0.7384	0.0984	0.0572	0.0940	0.1041	0.2981	0.2144	0.0732	0.0955	0.9176	0.2691	0.3064
V	0.6482	0.2761	0.7066	0.9371	0.7995	0.2365	0.1843	0.6082	0.7395	0.2948	0.5431	0.2702
ExR	0.9559	0.2111	0.0612	0.8488	0.8395	0.7908	0.1264	0.3828	0.1195	0.4729	0.4809	0.3497
ExB	0.7240	0.3610	0.6374	0.6980	0.4183	0.4662	0.1629	0.8326	0.3456	0.9791	0.5625	0.2525
ExG	0.7549	0.0972	0.0668	0.2196	0.1851	0.0373	0.7835	0.2897	0.2375	0.3225	0.2994	0.2645
MR	0.1122	0.3797	0.0316	0.3538	0.1319	0.2051	0.7779	0.3442	0.8124	0.2239	0.3373	0.2664
MG	0.1253	0.2337	0.7893	0.3044	0.7582	0.4657	0.6985	0.1806	0.6098	0.9258	0.5091	0.2858
MB	0.4920	0.3143	0.3254	0.4192	0.4458	0.2395	0.1536	0.2876	0.7695	0.8005	0.4247	0.2144
SDR	0.0136	0.1167	0.0552	0.1139	0.0781	0.3762	0.0172	0.2690	0.2002	0.2155	0.1456	0.1177
SDG	0.0108	0.0835	0.0851	0.0264	0.0883	0.3825	0.2080	0.0194	0.2743	0.0556	0.1234	0.1242
SDB	0.0012	0.0522	0.0459	0.0427	0.2194	0.2193	0.0584	0.3989	0.1289	0.0064	0.1173	0.1267
MH	0.5757	0.0351	0.1605	0.9111	0.1469	0.9223	0.0254	0.2682	0.1353	0.8616	0.4042	0.3742
MS	0.6523	0.3068	0.0442	0.0642	0.2147	0.2119	0.3596	0.8859	0.4064	0.6697	0.3816	0.2766
MV	0.0160	0.9502	0.0441	0.3472	0.5527	0.7841	0.6615	0.5698	0.7959	0.2517	0.4973	0.3217
SDH	0.1672	0.4067	0.1900	0.3916	0.4765	0.3160	0.0673	0.3128	0.3755	0.0816	0.2785	0.1430
SDS	0.0252	0.4144	0.2204	0.1460	0.1776	0.2288	0.1337	0.4976	0.5146	0.0228	0.2381	0.1795
SDV	0.6078	0.2135	0.0684	0.0993	0.1304	0.3065	0.0287	0.1040	0.3757	0.2373	0.2172	0.1757

6. Conclusion

In this paper, a new SSL graph-based approach is proposed to perform interactive image segmentation. It employs undirected and unweighted k NN graphs to propagate labels from nodes representing labeled pixels to nodes representing unlabeled pixels. Computer simulations with some real-world images show that the proposed approach is effective, achieving segmentation accuracy similar to those achieved by some state-of-the-art methods.

As future work, the method will be applied on more images and more features may be extracted. Methods to automatically define the parameters k and λ may also be explored. Graph generation may also be improved to provide further increase in segmentation accuracy.

Moreover, the proposed method works for multiple labels simultaneously at no extra cost, which is an interesting property not often exhibited by other interactive image segmentation methods. This feature will also be explored in future works.

Acknowledgment

The author would like to thank the São Paulo Research Foundation - FAPESP (grant #2016/05669-4) and the National Council of Technological and Scientific Development - CNPq (grant #475717/2013-9) for the financial support.

References

Artan, Y. (2011). Interactive image segmentation using machine learning techniques. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pages 264–269.

- Artan, Y. and Yetik, I. (2010). Improved random walker algorithm for image segmentation. In *Image Analysis Interpretation (SSIAI), 2010 IEEE Southwest Symposium on*, pages 89–92.
- Belkin, M., Matveeva, I., and Niyogi, P. (2004). Regularization and semisupervised learning on large graphs. In *Conference on Learning Theory*, pages 624–638. Springer.
- Belkin, M., P., N., and Sindhwani, V. (2005). On manifold regularization. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005)*, pages 17–24, New Jersey. Society for Artificial Intelligence and Statistics.
- Blake, A., Rother, C., Brown, M., Perez, P., and Torr, P. (2004). Interactive image segmentation using an adaptive gmmrf model. In Pajdla, T. and Matas, J., editors, *Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 428–441. Springer Berlin Heidelberg.
- Blum, A. and Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 19–26, San Francisco. Morgan Kaufmann.
- Boykov, Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112 vol.1.
- Breve, F., Quiles, M., and Zhao, L. (2015a). Interactive image segmentation of non-contiguous classes using particle competition and cooperation. In Gervasi, O., Murgante, B., Misra, S., Gavrilova, M. L., Rocha, A. M. A. C., Torre, C., Taniar, D., and Apduhan, B. O., editors, *Computational Science and Its Applications – ICCSA 2015*, volume 9155 of *Lecture Notes in Computer Science*, pages 203–216. Springer International Publishing.
- Breve, F., Quiles, M. G., and Zhao, L. (2015b). Interactive image segmentation using particle competition and cooperation. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.
- Breve, F. A. (2015). Auto feature weight for interactive image segmentation using particle competition and cooperation. In *Proceedings - XI Workshop de Visão Computacional WVC'2015*, pages 164–169.
- Chapelle, O., Schölkopf, B., and Zien, A., editors (2006). *Semi-Supervised Learning. Adaptive Computation and Machine Learning*. The MIT Press, Cambridge, MA.
- Ding, L. and Yilmaz, A. (2008). Image segmentation as learning on hypergraphs. In *Machine Learning and Applications, 2008. ICMLA '08. Seventh International Conference on*, pages 247–252.
- Ding, L. and Yilmaz, A. (2010). Interactive image segmentation using probabilistic hypergraphs. *Pattern Recognition*, 43(5):1863 – 1873.
- Ducournau, A. and Bretto, A. (2014). Random walks in directed hypergraphs and application to semi-supervised image segmentation. *Computer Vision and Image Understanding*, 120(0):91 – 102.

- Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226.
- Gonzalez, R. C. and Woods, R. E. (2008). *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Grady, L. (2006). Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1768–1783.
- Joachims, T. (2003). Transductive learning via spectral graph partitioning. In *Proceedings of International Conference on Machine Learning*, pages 290–297. AAAI Press.
- Li, J., Bioucas-Dias, J., and Plaza, A. (2010). Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(11):4085–4098.
- Lichman, M. (2013). UCI machine learning repository.
- Paiva, A. and Tasdizen, T. (2010). Fast semi-supervised image segmentation by novelty selection. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1054–1057.
- Protiere, A. and Sapiro, G. (2007). Interactive image segmentation via adaptive weighted distances. *Image Processing, IEEE Transactions on*, 16(4):1046–1057.
- Rother, C., Kolmogorov, V., and Blake, A. (2004). “grabcut”: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314.
- Shapiro, L. and Stockman, G. (2001). *Computer Vision*. Prentice Hall.
- Smith, A. R. (1978). Color gamut transform pairs. In *ACM Siggraph Computer Graphics*, volume 12, pages 12–19. ACM.
- Wang, F. and Zhang, C. (2008). Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):55–67.
- Xu, J., Chen, X., and Huang, X. (2008). Interactive image segmentation by semi-supervised learning ensemble. In *Knowledge Acquisition and Modeling, 2008. KAM '08. International Symposium on*, pages 645–648.
- Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, volume 16, pages 321–328. MIT Press.
- Zhu, X. (2005). Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.
- Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 912–919.

Um Levantamento do Uso de Aprendizado Profundo em Análise de Sentimentos

Eduardo de O. Andrade¹, José Viterbo¹, Cristina N. Vasconcelos¹

¹ Instituto de Computação
Universidade Federal Fluminense (UFF) – Niterói, RJ – Brazil

{eandrade, viterbo, crisnv}@ic.uff.br

Abstract. *Deep learning consists of a machine learning approach that has a high level of abstraction, being able to obtain good results with large databases through the use of increasingly advanced hardware such as Graphical Process Units (GPUs). The inclusion of sentiment analysis in the deep learning environment is one of the success cases. The textual preprocessing, unsupervised algorithms of word vectorization and neural networks, compose the steps for the classification of opinions. This article presents a survey on the subject with the main techniques, comparing the accuracy and macro-F1 obtained among several works in the area.*

Resumo. *O aprendizado profundo consiste em uma abordagem de aprendizado de máquina que possui elevado nível de abstração, sendo capaz de obter bons resultados com grandes bases de dados, através do emprego de hardwares cada vez mais avançados, como as Graphical Process Units (GPUs). A inclusão de análise de sentimentos no ambiente de aprendizado profundo é um dos casos de sucesso. O pré-processamento textual, algoritmos não supervisionados de vetorização de palavras e redes neurais, compõem as etapas para a classificação de opiniões. Este artigo apresenta um levantamento sobre o assunto com as principais técnicas, comparando a acurácia e macro-F1 obtidas entre diversos trabalhos na área.*

1. Introdução

O número de usuários com acesso à Internet não para de crescer, incluindo cerca de 40% da população mundial atual¹. Neste cenário, podemos acrescentar consequentemente o aumento do conteúdo disponibilizado na Web. Este conjunto de dados, incluindo grande parte dos dados em formato de texto, são alvo da área de estudo em Processamento de Linguagem Natural (PLN).

A análise de sentimentos (*sentiment analysis*, em inglês) abrange o estudo das opiniões, sentimentos, emoções e atitudes das pessoas em relação a entidades, como organizações, serviços, eventos, produtos, e seus atributos [Liu 2012]. Uma revisão sobre algumas técnicas de aprendizado de máquina (*machine learning*, em inglês) em análise de sentimentos já foi anteriormente elaborada pelo mesmo grupo de trabalho onde encontram-se os autores deste artigo [Lunardi et al. 2015].

¹<http://www.internetlivestats.com/internet-users/>

Quando tratamos de grandes bases de dados [Ward and Barker 2013], há uma subárea de aprendizado de máquina que está demonstrando bons resultados: o aprendizado profundo (*deep learning*, em inglês) [LeCun et al. 2015]. Como o próprio nome já diz, o aprendizado profundo apresenta diversos níveis de representação através de módulos não-lineares, nos quais os dados sofrem um processo de transformação, tornando-os mais abstratos conforme atingem níveis mais altos. Desta maneira, alguns detalhes dos dados acabam por apresentar maior relevância e outros detalhes menos importantes, são desconsiderados. Por fim, funções muito complexas podem ser aprendidas mais facilmente por este tipo de aprendizagem.

Estudos envolvendo aprendizado profundo e análise de sentimentos são recentes e por ser um assunto atual com constantes aprimoramentos, é necessária uma melhor avaliação sobre os detalhes que possam colaborar para melhorar os resultados desejados pelas empresas e pesquisadores, sendo este o objetivo deste trabalho.

A organização deste artigo segue com a Seção 2, no qual são vistos os trabalhos relacionados. A Seção 3 apresenta uma descrição sobre análise de sentimentos e aprendizado profundo. Depois disto, temos a Seção 4 que procura avaliar os algoritmos não supervisionados de vetorização de palavras e as arquiteturas de redes neurais mais importantes em aprendizado profundo e análise de sentimentos. Na Seção 5 temos o desempenho alcançado em diversos trabalhos e os algoritmos utilizados, comparando as acurácias e macro-F1 obtidas. Por fim, temos a Seção 6 com as conclusões.

2. Trabalhos Relacionados

Em [Rojas-Barahona 2016] é realizada uma revisão sobre o tema, abordando principalmente as arquiteturas de redes neurais de aprendizado profundo (inclusive arquiteturas híbridas) e suas aplicações em análise de sentimentos, demonstrando também alguns resultados para certas bases de dados após a utilização destas redes.

Apesar de tratar em [Filho 2016] o caso de análise de sentimentos através do aprendizado de máquina, é importante lembrar que o aprendizado profundo é uma subárea do aprendizado de máquina e sendo assim, diversas características são relacionadas. São observadas diferentes abordagens em *tweets*, textos com até 140 caracteres extraídos do *microblog* Twitter, incluindo técnicas de pré-processamento textual.

Outro estudo importante foi elaborado em [Erhan et al. 2010], no qual é mencionado a importância do pré-treinamento em aprendizado profundo e como isto é capaz de melhorar os resultados finais. São citados brevemente os *autoencoders*, que são redes neurais treinadas para reconstruir as entradas originais, mas procurando extrair as características mais relevantes através da redução da dimensionalidade.

Este artigo lida com o passo a passo para a análise de sentimentos com ênfase no aprendizado profundo, avaliando as escolhas possíveis, diferentemente dos outros trabalhos. As comparações entre os algoritmos na Seção 5 são feitas através das publicações mais recentes na área, também colaborando com os caminhos que podem ser selecionados.

3. Análise de Sentimentos e Aprendizado Profundo

Nesta seção são descritas brevemente as áreas de análise de sentimentos e aprendizado profundo. A Seção 3.1 é relativa à análise de sentimentos e a Seção 3.2 ao apren-

dizado profundo.

3.1. Análise de Sentimentos

As opiniões públicas em geral sempre foram de suma importância para diversas entidades, como empresas e governos. A avaliação de um determinado produto pelos clientes é fator crucial para seu sucesso ou fracasso. O que as pessoas pensam sobre os próximos candidatos nas eleições também. Estes são exemplos básicos, nos quais a análise de sentimentos pode apresentar enorme influência.

Para isto, a análise de sentimentos pode contribuir com o estudo de polaridade das opiniões. Resumidamente, as opiniões públicas com o advento das redes sociais, principalmente, estão expressas de uma forma que podem ser “capturadas” por outros usuários e então, analisadas. A “captura” destas opiniões serve de entrada para algoritmos que procuram extrair as polaridades dos textos.

Antes da etapa de pré-treinamento, existe o pré-processamento de texto. O pré-processamento é fundamental para colaborar com a execução dos algoritmos não supervisionados que lidam com dados não rotulados e agrupamentos (*clusters*, em inglês) [Hemalatha et al. 2012]. Algumas técnicas de pré-processamento textual destacam-se [Filho 2016] e são mais utilizadas que outras. Apesar disto, há discussões acerca da eficiência destas técnicas [Jianqiang 2015]. Portanto, os resultados podem diferir de acordo com as técnicas e bases de dados selecionadas.

3.2. Aprendizado profundo

Podemos dizer que as múltiplas camadas presentes nas redes neurais, que são arquiteturas características do aprendizado profundo, procuram emular o funcionamento de um cérebro humano, chamando geralmente de neurônios o que seriam os nós destas redes. As redes neurais são capazes de aprender com os dados e tomar decisões.

Uma rede neural alimentada para frente (*feedforward neural network*, em inglês) [Bebis and Georgiopoulos 1994] é a arquitetura mais simples que podemos imaginar em termos de redes neurais em aprendizado profundo e pode ser observada na Figura 1.

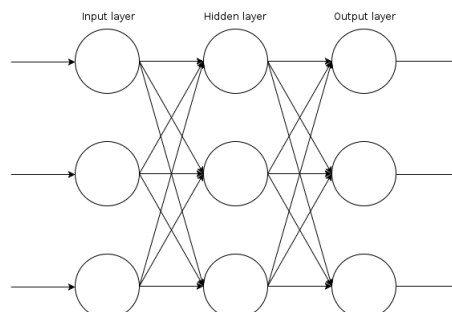


Figura 1. Representação de uma *feedforward neural network* com apenas três camadas

Em sua forma mais reduzida, esta arquitetura é representada por uma camada de entrada (*input layer*, em inglês), uma camada escondida (*hidden layer*, em inglês)

e uma camada de saída (*output layer*, em inglês) e não possui ciclos. Dado um vetor de entradas \mathbf{i} (representado pelos valores atribuídos aos neurônios da *input layer*), a matriz de pesos \mathbf{P}_{ih} (representada como um conjunto de transformações que ocorrem nos neurônios da *input layer* com destino aos neurônios da *hidden layer*) e a função de ativação f_h (responsável pela ativação dos neurônios da *hidden layer*), temos a equação 1 que representa o vetor da *hidden layer* \mathbf{h} .

$$\mathbf{h} = f_h(\mathbf{i}P_{ih}) \quad (1)$$

Seguindo o raciocínio anterior, dado um vetor de entradas \mathbf{h} (representado pelos valores atribuídos aos neurônios da *hidden layer*), a matriz de pesos \mathbf{P}_{ho} (representada como um conjunto de transformações que ocorrem nos neurônios da *hidden layer* com destino aos neurônios da *output layer*) e a função de ativação f_o (responsável pela ativação dos neurônios da *output layer*), temos a equação 2 que representa o vetor da *output layer* \mathbf{o} .

$$\mathbf{o} = f_o(\mathbf{h}P_{ho}) \quad (2)$$

Sobre o aprendizado profundo, o mais “correto” para uma rede neural ser considerada profunda é a implementação de múltiplas *hidden layers* e o aumento de complexidade da rede. Fora a abordagem não supervisionada, os algoritmos de aprendizado profundo também são capazes de lidar com as abordagens supervisionada e semi-supervisionada que trabalham com conjunto de dados rotulados, ou seja, com suas classes predefinidas mesmo que parcialmente, como pode ocorrer no caso semi-supervisionado.

4. Técnicas Empregadas em Aprendizado Profundo

Aqui temos na Seção 4.1 a avaliação dos algoritmos não supervisionados que abordam a vetorização de palavras. Após isto, na Seção 4.2 algumas arquiteturas de redes neurais aplicadas à análise de sentimentos são discutidas.

4.1. Algoritmos Não Supervisionados de Vetorização de Palavras

Este passo chamado de pré-treinamento pode ser realizado após o pré-processamento textual. Abordaremos nesta seção os três algoritmos mais relevantes: word2vec (Seção 4.1.1), GloVe (Seção 4.1.2) e SSWE (Seção 4.1.3).

4.1.1. word2vec

O word2vec foi introduzido em [Mikolov et al. 2013] como um algoritmo de “incorporação de palavras” (*word embedding*, em inglês), ou seja, um modelo de aprendizagem de representações vetoriais de palavras. Quando lidamos com PLN, as palavras são geralmente tratadas como símbolos atômicos discretos. É mais difícil diferenciar um carro de uma bicicleta apenas considerando as palavras *car* e *bike* do que duas imagens distintas destes objetos, por exemplo.

A arquitetura do word2vec é similar a uma rede neural, no qual o vetor de entrada corresponde a um vocabulário de n posições. O vocabulário consiste em todas as palavras que aparecem no texto que estamos considerando ou nas palavras que apareçam pelo menos um determinado número de vezes no mínimo. Logo, o vetor de entradas é representado por n = número de palavras presente no vocabulário.

O objetivo do word2vec é aprender sobre palavras que aparecem em contextos similares. No exemplo das palavras *car* e *bike*, o algoritmo poderia ajudar na classificação de um texto na categoria meios de transporte. Em análise de sentimentos, o word2vec colabora com a classificação entre as polaridades possíveis de um texto.

4.1.2. GloVe

GloVe [Pennington et al. 2014] é uma abreviação para *Global Vectors*, sendo um modelo baseado em contagem, diferente do word2vec que é um modelo preditivo. A implementação do GloVe foi uma tentativa de utilizar as vantagens dos métodos de contagem e extração linear de significados.

O algoritmo é capaz de capturar diretamente as estatísticas de um texto inteiro. Na Tabela 1 podemos visualizar uma exemplificação de como funciona o GloVe para o caso de *car* e *bike*, demonstrando a frequência entre as palavras presentes nas linhas e colunas dentro de um mesmo contexto.

Tabela 1. Coocorrência entre palavras considerando os exemplos *car* e *bike*

Probabilidade	x = engine	x = handlebar	x = wheel	x = princess
P (x car)	grande	pequena	grande	pequena
P (x bike)	pequena	grande	grande	pequena
P (x car) / P (x bike)	grande	pequena	~1	~1

Como podemos observar, a Tabela 1 é formulada através da implementação de uma matriz de coocorrência, no qual cada entidade em uma linha (palavras) é pareada com as entidades presentes nas colunas (contexto). No caso, nota-se que as palavras *wheel* e *princess* não são boas para discriminar *car* e *bike*, diferentemente de *engine* e *handlebar*.

Em comparação com o word2vec, o GloVe apresenta-se como um algoritmo mais simples de ser paralelizado, permitindo que mais dados sejam facilmente treinados. Os autores do GloVe demonstram que considerando um mesmo texto, vocabulário, tamanho de janela e tempo de treinamento, o GloVe consegue superar a performance do word2vec.

4.1.3. SSWE

O SSWE (*Sentiment-Specific Word Embedding*, em inglês) foi introduzido como uma adaptação da rede neural apresentada em C&W [Collobert et al. 2011] na procura pelo aprendizado de sentimentos nas *word embeddings* para aprimorar os resultados do word2vec na presença de *emojicons* em *tweets* [Tang et al. 2014b].

O SSWE procura capturar os sentimentos das sentenças e o contexto sintático das palavras. Isto ocorre através da predição de um vetor de duas dimensões para cada

n -gram que serve como entrada, no qual dois escalares são responsáveis por representar o “valor de linguagem” e o “valor de sentimento”.

Em relação ao word2vec, no trabalho original do SSWE é apresentada uma tabela comparativa com algumas bases de dados em que o SSWE apresenta uma performance superior. Porém, quando tratamos da utilização deste algoritmo na literatura, nota-se uma adesão maior logo após a sua publicação, mas com o decorrer do tempo o word2vec e GloVe continuaram sendo mais adotados pelos autores, talvez pela característica original do SSWE em trabalhar com *emoticons* em *tweets*.

4.2. Arquiteturas de Redes Neurais

Após a execução de um algoritmo não supervisionado de vetorização de palavras, uma possibilidade é a inserção das saídas (*word embeddings*) como entrada para uma rede neural com abordagem supervisionada ou semi-supervisionada. Representações contínuas em um espaço no qual as palavras de significados similares são próximas uma das outras podem ser aprendidas por estas redes neurais, inclusive em situações envolvendo palavras de difícil reconhecimento. Nesta seção encontram-se as arquiteturas mais importantes em análise de sentimentos: rede neural de convolução (Seção 4.2.1), rede neural recorrente (Seção 4.2.2) e rede neural recursiva (Seção 4.2.3).

4.2.1. Rede Neural de Convolução

Originalmente aplicada em outras áreas, como visão computacional e reconhecimento de fala, a rede neural de convolução (*Convolutional Neural Network* ou CNN, em inglês) também é capaz de apresentar bons resultados em análise de sentimentos [Kim 2014]. Basicamente, esta rede neural consiste em uma *feedforward neural network* baseada no córtex visual animal, no qual há a presença de diversas camadas, que se dividem entre camadas de convolução, agrupamento e completamente conectada (*convolutional layers*, *pooling* e *fully connected* respectivamente, em inglês).

As sentenças podem ser representadas como uma matriz. Cada linha desta matriz corresponderá tipicamente a uma palavra, ou seja, cada linha é formada por um vetor que representa uma palavra. Tipicamente, a largura dos filtros aplicados na camada de convolução, que normalmente é a camada posterior a camada de entrada, são do mesmo tamanho da matriz de entrada. Como saída deste tipo de camada, temos uma “característica convolucionada” (*convolved feature*, em inglês). Com este filtro, algumas características das palavras de entrada podem ser destacadas.

A camada de agrupamento que segue, tem como maneira mais comum de agrupar, uma operação de *max* para cada filtro. O agrupamento pode ocorrer apenas sobre uma janela, mas em PLN costuma-se aplicar sobre a matriz inteira, produzindo apenas um número para cada filtro. Esta ação retorna uma saída com uma matriz de tamanho fixo, o que é importante para a classificação em análise de sentimentos, auxiliando para sentenças de tamanhos variados e filtros também, retornando saídas com as mesmas dimensões para classificação. Assim como na camada de convolução, a camada de agrupamento também tem sua função que é a manutenção de informações mais relevantes. Cada filtro pode ser pensado como um “detector” de característica e cada operação de *max* como um “realçador”.

A camada completamente conectada recebe todas as saídas provenientes dos neurônios da camada anterior e conecta em cada neurônio presente em sua camada. Importante ressaltar que não há nenhum tipo de camada de convolução ou agrupamento após uma camada completamente conectada. As características aqui representadas são de alto nível e aprendidas através de combinações não lineares. No final utiliza-se uma função de *softmax* para classificação. Um argumento a favor das CNNs é a rapidez. Elas são implementadas no nível de *hardware* das GPUs. Fora isto, os filtros de convolução conseguem aprender boas representações de forma automática, sem necessidade de representar todo o vocabulário.

4.2.2. Rede Neural Recorrente

A principal característica de uma rede neural recorrente (*Recurrent Neural Networks* ou RNNs, em inglês) é a formação de ciclos diretos entre seus neurônios. Isto permite com que a rede obtenha um processamento temporal e aprenda sobre sequências através da criação de estados internos. A complexidade das RNNs é geralmente maior do que nas CNNs e também demandam uma capacidade computacional superior, além de mais tempo para sua execução.

A existência de estado nas RNNs faz com que elas mantenham parte da ativação das entradas anteriores e “realimentem” os cálculos da rede a cada passo de tempo. Definindo t como o t -ésimo passo de tempo, $\mathbf{i}(t)$ e $\mathbf{o}(t)$ como os vetores de entrada e saída respectivamente, $\mathbf{h}(t)$ como o conjunto de ativações (estado) da *hidden layer*, f_h e f_o como as funções de ativação da *hidden layer* e *output layer* respectivamente, \mathbf{P}_{ih} como a matriz de peso da *input layer* com destino a *hidden layer*, \mathbf{P}_{pp} como a matriz de peso da *hidden layer* com destino a própria *hidden layer* e \mathbf{P}_{ho} como a matriz de peso da *hidden layer* com destino a *output layer*, temos as equações 3 e 4.

$$\mathbf{h}(t) = f_h(\mathbf{i}(t)\mathbf{P}_{ih} + \mathbf{h}(t-1)\mathbf{P}_{pp}) \quad (3)$$

$$\mathbf{o}(t) = f_o(\mathbf{h}(t)\mathbf{P}_{ho}) \quad (4)$$

A rede neural *Long Short Term Memory* (LSTM) é um tipo muito bem-sucedido de RNN. A implementação da LSTM foi uma tentativa de resolver o problema de “dependência de longo prazo” [Sundermeyer et al. 2012]. Isto ocorreu com as células de memórias presentes nas LSTMs que são capazes de preservar estados diante de longos prazos de tempo.

4.2.3. Rede Neural Recursiva

A rede neural recursiva (*Recursive Neural Network* ou ReNN, em inglês) assim como as RNNs, pode produzir uma saída através de uma entrada de tamanho variável. Nas ReNNs o mesmo conjunto de pesos são aplicados de forma recursiva para realizar uma predição. Os algoritmos das ReNNs seguem uma topologia de árvore e costumam executar de modo *bottom-up*, nos quais as *word embeddings* são utilizadas como parâme-

tros e computadas pelos nós pais até o topo e servem como *features* para um classificador *softmax* [Socher et al. 2013]. Para cada passo até o topo da árvore, são avaliadas as polaridades de cada palavra até a classificação final.

A performance entre uma RNN e uma ReNN é parecida, variando de acordo com a tarefa especificada. A diferença está no reuso possível das estruturas no processo de classificação, não sendo preciso considerar uma fusão de todas as palavras até certo ponto de tempo, caso que ocorre nas RNNs.

A ReNN procura lidar com continuidade. Logo, quando se trata de representações de palavras nas quais há muitas lacunas, a saída da ReNN será ruim pois o algoritmo não conseguirá trabalhar corretamente com uma estrutura contínua e o sentido final poderá não ser o esperado.

5. Avaliação Comparativa

Na Tabela 2 podem ser observados os valores de acurácia obtidos na base de dados sobre revisões de filmes, *Sentiment Treebank*². Esta base é composta de 215.154 frases e 11.855 sentenças. A tendência dos trabalhos atualmente está na utilização de RNNs (principalmente LSTMs), nos quais são obtidos resultados cada vez melhores, apesar de muitos também utilizarem as CNNs. Os algoritmos não supervisionados de vetorização de palavras que predominam são o word2vec e GloVe. Entre parêntesis na coluna “Rede neural” temos os nomes dados aos modelos modificados de tipos existentes de redes neurais no trabalho de cada autor. As comparações binária (positivo e negativo) e multiclasse (positivo, um pouco positivo, neutro, um pouco negativo e negativo) foram realizadas na Tabela 2.

Os maiores valores de acurácia obtidos pelos algoritmos tradicionais de aprendizado de máquina na Tabela 2 ocorreram no trabalho dos autores do paragraph2vec [Le and Mikolov 2014], implementado em conjunto com uma regressão logística. Na Figura 2 observa-se o gráfico da evolução dos percentuais de acurácia obtidos na base de dados *Sentiment Treebank* de acordo com a Tabela 2.

Tabela 2. Comparação dos trabalhos de aprendizado profundo e aprendizado de máquina em análise de sentimentos no domínio de revisões de filmes

Autores	Classificação	Algoritmo Não Supervisionado de Vetorização de Palavras	Rede Neural e Acurácia (%)	Algoritmo de Aprendizado de Máquina, Autores e Acurácia (%)	Ganho (%)
[Socher et al. 2013]	Binária	Inicialização aleatória através de uma distribuição uniforme	ReNN (RNTN) 85.4	Naive Bayes with bag of bigram features (biNB) [Socher et al. 2013] 83.1	+2.3
	Multiclasse		ReNN (RNTN) 45.7	Naive Bayes with bag of bigram features (biNB) [Socher et al. 2013] 41.9	+3.8
[Kim 2014]	Binária	word2vec	CNN (CNN-multichannel) 88.1	Regressão Logística [Le and Mikolov 2014] 87.8	+0.3
	Multiclasse		CNN (CNN-non-static) 48.0	Regressão Logística [Le and Mikolov 2014] 48.7	-0.7
[Kumar et al. 2015]	Binária	GloVe	RNN (DMN) 88.6	Regressão Logística [Le and Mikolov 2014] 87.8	+0.8
	Multiclasse		RNN (DMN) 52.1	Regressão Logística [Le and Mikolov 2014] 48.7	+3.4
[Munkhdalai and Yu 2016]	Binária	GloVe	RNN (NSE) 89.7	Regressão Logística [Le and Mikolov 2014] 87.8	+1.9
	Multiclasse		RNN (NSE) 52.8	Regressão Logística [Le and Mikolov 2014] 48.7	+4.1
[Looks et al. 2017]	Binária	GloVe	RNN (Tree-LSTM) 90.2	Regressão Logística [Le and Mikolov 2014] 87.8	+2.4
	Multiclasse		RNN (Tree-LSTM) 53.6	Regressão Logística [Le and Mikolov 2014] 48.7	+4.9
[Radford et al. 2017]	Binária	word2vec	RNN (Byte mLSTM) 91.8	Regressão Logística [Le and Mikolov 2014] 87.8	+4.0
	Multiclasse		RNN (Byte mLSTM) 52.9	Regressão Logística [Le and Mikolov 2014] 48.7	+4.2

²<https://nlp.stanford.edu/sentiment/index.html>

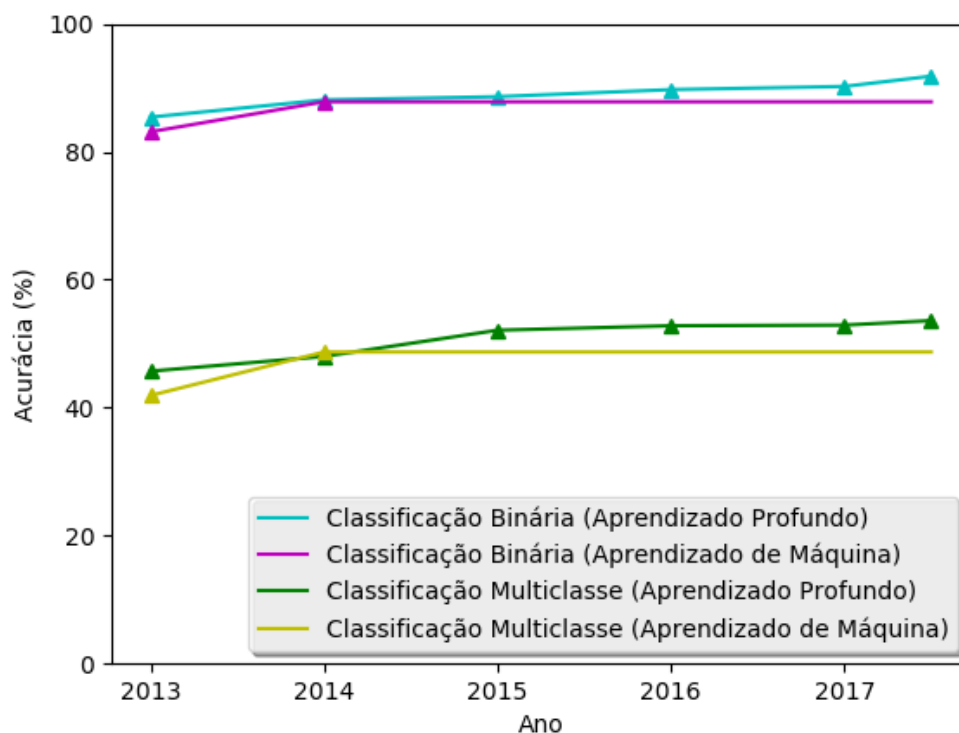


Figura 2. Performance de aprendizado profundo e aprendizado de máquina ao longo dos anos no domínio de revisões de filmes

O SemEval (*Semantic Evaluation*, em inglês) é um evento que ocorre anualmente e avalia sistemas computacionais de análise semântica. A Twitter2014³, Twitter2016⁴ e Twitter2017⁵ (Twitter2017 foi um nome dado pelos autores deste trabalho para a base de dados de teste utilizada no ano de 2017 mas não é o nome original) são algumas das bases de dados de testes que foram utilizadas. A Tabela 3 apresenta os valores de macro-F1 obtidos nestas bases, sendo que a Twitter2014 é composta de 982 *tweets* positivos, 669 neutros e 202 negativos, totalizando 1853 *tweets*. No caso da Twitter2016, são 7059 *tweets* positivos, 10342 neutros e 3231 negativos, totalizando 20632 *tweets*. Por último, a Twitter2017 possui 2375 *tweets* positivos, 5937 neutros e 3972 negativos, totalizando 12284 *tweets*. As CNNs estão sempre presentes entre os tipos de redes neurais mais utilizadas e com maiores valores de macro-F1 obtidos. Entre os algoritmos não supervisionados de vetorização de palavras temos uma grande variedade na adoção. As comparações realizadas na Tabela 3 foram multiclasse, entre as classes positivo, neutro e negativo.

Os ganhos das redes neurais de aprendizado profundo comparados aos algoritmos tradicionais de aprendizado de máquina nestas bases formadas por *tweets* podem ser vistos na Figura 3 e estão de acordo com a Tabela 3.

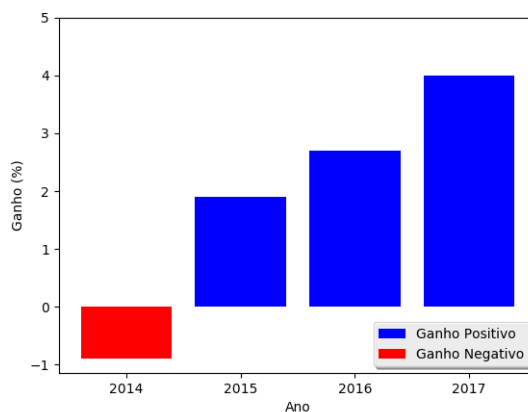
³<http://alt.qcri.org/semeval2014/task9/index.php?id=data-and-tools>

⁴<http://alt.qcri.org/semeval2016/task4/index.php?id=data-and-tools>

⁵<http://alt.qcri.org/semeval2017/task4/index.php?id=data-and-tools>

Tabela 3. Comparação dos trabalhos de aprendizado profundo e aprendizado de máquina em análise de sentimentos no domínio de tweets (geral)

Autores	Base de Dados	Algoritmo Não Supervisionado de Vetorização de Palavras	Rede Neural e Macro-F1 (%)	Algoritmo de Aprendizado de Máquina, Autores e Macro-F1 (%)	Ganho (%)
[Tang et al. 2014a]	Twitter2014	SSWE	CNN 70.1	Regressão Logística [Miura et al. 2014] 71.0	-0.9
[Dong et al. 2015]	Twitter2014	word2vec	CNN 74.4	<i>Support Vectors Regression (SVR)</i> e <i>Radial Basis Function (RBF)</i> [Astudillo et al. 2015] 72.5	+1.9
[Deriu et al. 2016]	Twitter2016	GloVe	CNN 63.3	SVM [Giorgis et al. 2016] 60.5	+2.7
[Cliche 2017]	Twitter2017	word2vec, fastText [Joulin et al. 2016] e GloVe	CNN e RNN (LSTM) 68.5	SVM e Algoritmo Genético [Miranda-Jiménez et al. 2017] 64.5	+4.0

**Figura 3. Ganho de aprendizado profundo em relação ao aprendizado de máquina ao longo dos anos no domínio de tweets (geral)**

6. Conclusão

Foram vistas diferentes abordagens de aprendizado profundo em análise de sentimentos neste trabalho, procurando descrever os principais pontos envolvidos no assunto. Avaliou-se o desempenho através de algoritmos que foram executados em dois domínios distintos (revisões de filmes e geral), apresentando as métricas de acurácia e macro-F1 obtidas. Com isto, procurou-se demonstrar caminhos que podem ser seguidos para a execução das redes neurais e classificação de opiniões.

Como trabalho futuro, os autores deste artigo pretendem utilizar uma CNN com modificações próprias em outras bases de dados. Esta opção está de acordo com as requisições de *hardware* e velocidade das CNNs em comparação com os outros tipos de redes neurais. Em relação aos algoritmos não supervisionados de vetorização de palavras, são precisos alguns testes para a melhor escolha, apesar da preferência pelo GloVe, já que isto pode incluir *word embeddings* pré-processadas e/ou treinamentos realizados em grandes bases de dados, demandando muito tempo. Assim, objetiva-se colaborar com o cenário vigente, trazendo novos resultados e questionamentos. Alguns trabalhos atuais e interessantes podem ser vistos em outros meios como mercado financeiro⁶ e detecção de personalidade [Majumder et al. 2017].

Referências

Astudillo, R. F., Amir, S., Ling, W., Martins, B., Silva, M., Trancoso, I., and Redol, R. A. (2015). Inesc-id: A regression model for large scale twitter sentiment lexicon in-

⁶<http://wangd.csl.t.org/talks/pdf/SentimentAnalysisinFinanceMarketForecast.pdf>

- duction. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 613–618.
- Bebis, G. and Georgiopoulos, M. (1994). Feed-forward neural networks. *IEEE Potentials*, 13(4):27–31.
- Cliche, M. (2017). Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. *arXiv preprint arXiv:1704.06125*.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.
- Deriu, J., Gonzenbach, M., Uzdilli, F., Lucchi, A., De Luca, V., and Jaggi, M. (2016). Swis-scheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. *Proceedings of SemEval*, pages 1124–1128.
- Dong, L., Wei, F., Yin, Y., Zhou, M., and Xu, K. (2015). Splusplus: A feature-rich two-stage classifier for sentiment analysis of tweets. *SemEval-2015*, page 515.
- Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660.
- Filho, C. A. F. (2016). Mineração de opiniões: um classificador ternário ou dois binários? Master’s thesis, Universidade Federal Fluminense.
- Giorgis, S., Rousas, A., Pavlopoulos, J., Malakasiotis, P., and Androutsopoulos, I. (2016). aueb. twitter. sentiment at semeval-2016 task 4: A weighted ensemble of svms for twitter sentiment analysis. *Proceedings of SemEval*, pages 96–99.
- Hemalatha, I., Varma, G. S., and Govardhan, A. (2012). Preprocessing the informal text for efficient sentiment analysis. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 1(2):58–61.
- Jianqiang, Z. (2015). Pre-processing boosting twitter sentiment analysis? In *Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on*, pages 748–753. IEEE.
- Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.
- Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.
- Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., and Socher, R. (2015). Ask me anything: Dynamic memory networks for natural language processing. *arXiv:1506.07285*.
- Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

- Looks, M., Herreshoff, M., Hutchins, D., and Norvig, P. (2017). Deep learning with dynamic computation graphs. *arXiv preprint arXiv:1702.02181*.
- Lunardi, A., Viterbo, J., and Bernardini, F. (2015). Um levantamento do uso de algoritmos de aprendizado supervisionado em mineração de opiniões. *Proceedings of XII Encontro Nacional de Inteligência Artificial e Computacional-ENIAC*, pages 262–269.
- Majumder, N., Poria, S., Gelbukh, A., and Cambria, E. (2017). Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32(2):74–79.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Miranda-Jiménez, S., Graff, M., and Tellez, E. S. (2017). Ingeotec at semeval 2017 task 4: A b4msa ensemble based on genetic programming for twitter sentiment analysis. *Proceedings of SemEval*.
- Miura, Y., Sakaki, S., Hattori, K., and Ohkuma, T. (2014). Teamx: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 628–632.
- Munkhdalai, T. and Yu, H. (2016). Neural semantic encoders. *arXiv preprint arXiv:1607.04315*.
- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.
- Radford, A., Jozefowicz, R., and Sutskever, I. (2017). Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.
- Rojas-Barahona, L. M. (2016). Deep learning for sentiment analysis. *Language and Linguistics Compass*, 10(12):701–719.
- Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., Potts, C., et al. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.
- Sundermeyer, M., Schlüter, R., and Ney, H. (2012). Lstm neural networks for language modeling. In *Interspeech*, pages 194–197.
- Tang, D., Wei, F., Qin, B., Liu, T., and Zhou, M. (2014a). Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 208–212.
- Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014b). Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565.
- Ward, J. S. and Barker, A. (2013). Undefined by data: a survey of big data definitions. *arXiv preprint arXiv:1309.5821*.

Aumentando o poder preditivo de classificadores lineares através de particionamento por classe

Nahim A. Souza, Tiago A. Almeida, Tiemi C. Sakata

¹ Departamento de Computação – Universidade Federal de São Carlos (UFSCar)
Rod. João Leme dos Santos, Km 110 – Bairro do Itinga - Sorocaba, SP - Brasil

nahimsouza@outlook.com, {talmeida,tiemi}@ufscar.br

Abstract. *This work describes a new approach to achieving good classification performances using linear prediction methods, even on datasets with non-linear separable data. The proposed approach, inspired by the division-and-conquer strategy, applies a clustering method to each subset of samples of the same class, defining groups by class. Subsequently, the intersection of these groups is calculated, creating a single partition, where each group can contain linearly separable samples. Then, one or more linear classifiers are trained. Experiments performed using datasets with different structural and complexity level indicate the overall performance of the prediction is similar or superior to well-known non-linear classification methods. The main advantages of this approach are (i) the need for much less effort and computational resources, especially in large databases, and (ii) the possibility of parallelization.*

Resumo. *Este trabalho descreve uma nova abordagem para obter bons desempenhos de classificação usando métodos de predição lineares, mesmo em bases com dados não separáveis linearmente. A abordagem proposta, inspirada na estratégia de divisão-e-conquista, aplica um método de agrupamento em cada subconjunto de amostras de mesma classe, definindo grupos por classe. Posteriormente, a interseção desses grupos é realizada, criando uma única partição, onde cada grupo pode conter amostras separáveis linearmente. Em seguida, um ou mais classificadores lineares são treinados. Experimentos realizados com bases de dados que possuem diferentes estruturas e níveis de complexidade demonstram que o desempenho geral da predição é muitas vezes semelhante ou superior aos métodos de classificação não-lineares. As principais vantagens dessa abordagem são (i) a necessidade de muito menor esforço e recursos computacionais, principalmente em bases de dados de grande porte, e (ii) a possibilidade de paralelização.*

1. Introdução

Nos últimos anos, diversos trabalhos têm surgido com o objetivo de oferecer métodos de classificação cada vez mais robustos para lidar com bases de dados complexas. A adoção de métodos de classificação não-lineares, tal como *deep learning*, vem tornando-se uma tendência, pois eles normalmente conseguem gerar modelos de predição extremamente sofisticados, robustos e com alto poder preditivo. Todavia, a desvantagem desses métodos é que a construção do modelo pode demandar muitos recursos computacionais e tempo de processamento [Britto et al. 2014].

No mesmo cenário de classificação, outras abordagens também têm recebido destaque nesse período, como por exemplo, a fusão e a seleção de classificadores [Britto et al. 2014, Kuncheva 2000, Almeida and Galvão 2016]. Muitas delas são baseadas no

princípio de divisão-e-conquista, através do particionamento da base de dados em vários grupos e do treinamento de classificadores especializados para cada um deles. Nesse escopo, [Almeida and Galvão 2016] demonstraram que a combinação de vários métodos de classificação “fracos”, por meio de um *ensemble*, pode ser uma boa solução para aumentar consideravelmente o poder preditivo da classificação sem necessariamente aumentar muito o esforço computacional.

Em geral, para uma mesma base de dados, métodos de classificação lineares são treinados muito mais rapidamente do que métodos não-lineares [Joachims 2006], pois possuem uma complexidade menor em relação ao número de operações realizadas [Zhouyu Fu et al. 2010, Fradkin 2008]. Consequentemente, a implementação de métodos lineares é mais simples e costuma ser menos custosa computacionalmente. Por outro lado, dependendo das características da base de dados, um modelo de predição linear pode apresentar desempenho insatisfatório na tarefa de classificação, pois as amostras podem ser inseparáveis linearmente. Neste cenário, este trabalho assume a hipótese de que, em alguns casos, é possível obter uma combinação de métodos lineares capaz de obter desempenho próximo ou superior a dos métodos de classificação não-lineares.

Para ilustrar, a Figura 1(a) apresenta uma base de dados com duas classes (verde e amarelo). Nenhum classificador linear seria capaz de separar corretamente as amostras das diferentes classes. Provavelmente, um classificador linear qualquer geraria uma hipótese ilustrada pela Figura 1(b). Neste caso, cerca de metade das amostras seria classificada corretamente. Entretanto, separando essas mesmas amostras em dois grupos, seria possível treinar dois classificadores lineares que, combinados, poderiam ser capazes de classificar corretamente um número muito maior de amostras (Figura 1(c)).

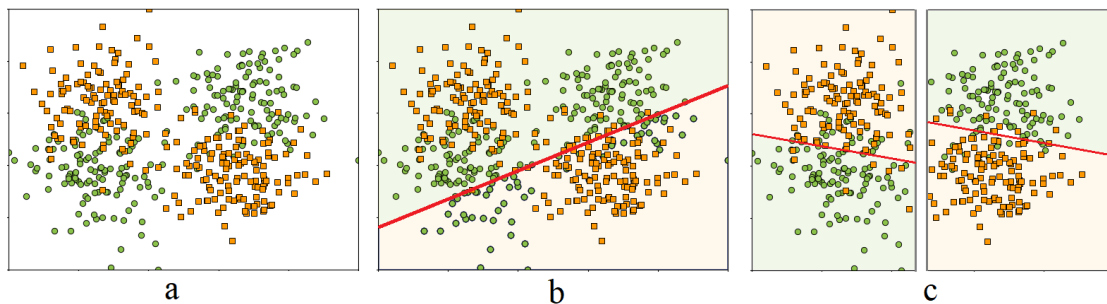


Figura 1. (a) Representação de uma base de dados com amostras de duas classes não separáveis linearmente. (b) A linha vermelha ilustra um classificador linear treinado sobre a base original. (c) As duas linhas vermelhas ilustram dois classificadores lineares obtidos a partir da divisão da base em dois grupos.

Este trabalho propõe uma abordagem baseada na estratégia de divisão-e-conquista para encontrar grupos compostos por amostras que tenham maior possibilidade de serem linearmente separáveis. Inicialmente, as amostras de uma mesma classe são particionadas e os grupos das diferentes classes tornam-se candidatos à união. Nesta etapa, os grupos com amostras muito próximas ou sobrepostas são mesclados para construir uma partição única. Em seguida, para cada grupo da partição final, um classificador linear é treinado e, posteriormente, combinado com os demais com o intuito de obter melhor poder preditivo na tarefa de classificação.

Uma das vantagens dessa abordagem está no fato de que a formação desses subconjuntos menores permite que a etapa de treinamento seja muito menos custosa dentro de cada grupo. Com isso, como será demonstrado, é possível construir um classificador

robusto, a partir de classificadores lineares, com uma eficiência superior ou equivalente à obtida pelos métodos de classificação não-lineares. Além disso, é possível que a etapa de treinamento seja totalmente paralelizada, permitindo um enorme ganho de desempenho [Fradkin 2008, Zhouyu Fu et al. 2010].

O restante deste trabalho está dividido da seguinte forma: a Seção 2 apresenta os principais trabalhos relacionados. Na Seção 3, a abordagem proposta é detalhada. Os experimentos e resultados são apresentados na Seção 4. Por fim, a Seção 5 conclui o trabalho e apresenta direcionamentos para trabalhos futuros.

2. Trabalhos relacionados

A combinação de métodos que oferecem aprendizado supervisionado e não-supervisionado para melhorar o desempenho na tarefa de classificação não é uma ideia nova. Nos últimos anos, diversos trabalhos têm aperfeiçoado essa ideia e mostrado que ela pode trazer resultados promissores nos mais diversos contextos [Jacobs et al. 1991, Kuncheva 2000, Chih-Wei Hsu and Chih-Jen Lin 2002, Kyriakopoulou and Kalam-boukis 2008, Coletta et al. 2014]. Uma das principais motivações reside no fato de que, a construção de um classificador único e com alto poder preditivo para uma base de dados complexa, geralmente é computacionalmente custosa [Britto et al. 2014, Almeida and Galvão 2016].

O particionamento da base de dados e o tratamento dos seus subconjuntos como instâncias menores do problema de classificação são exemplos de uso da abordagem de divisão-e-conquista [Zhouyu Fu et al. 2010, Fradkin 2008, Britto et al. 2014, Almeida and Galvão 2016]. O trabalho de [Kuncheva 2000], por exemplo, demonstrou que um algoritmo de agrupamento pode ser utilizado para produzir uma partição da base dados, onde podem ser gerados classificadores especializados para cada grupo. Essa abordagem apresentou bons resultados e foi adotada em diversas outras pesquisas subsequentes que, a partir dos classificadores especializados, propuseram métodos de seleção dinâmica [Britto et al. 2014, Kyriakopoulou and Kalam-boukis 2008] e *ensembles* [Ruta and Gabrys 2005] para construir soluções robustas.

[Fradkin 2008] observou que a maior parte das propostas presentes na literatura não considera que o rótulo das amostras pode ser uma informação útil para gerar grupos melhores na etapa de agrupamento, ou seja, grupos com características que podem ser úteis na etapa de classificação. Com isso em mente, ele aplicou agrupamento dentro de cada conjunto de amostras separadas por classe. Essa proposta trouxe vantagens em relação a ideia de [Kuncheva 2000], visto que a aplicação do agrupamento interno nas classes, usando o algoritmo k -médias, permitiu uma melhor identificação da estrutura das classes a partir da geração de grupos convexos [Fradkin 2008].

Para realizar as etapas de treinamento e classificação dos dados, [Fradkin 2008] emprega grupos gerados na etapa de agrupamento para rotular as amostras. A partir disso, para K classes, com k grupos gerados em cada uma delas, um único classificador R é treinado para toda a base de dados, de modo que ele seja capaz de identificar todos os $k \times K$ novos rótulos. Seus experimentos demonstraram que esses rótulos foram identificados com maior acurácia pelos classificadores lineares do que com os rótulos originais, melhorando, portanto, o desempenho geral da classificação.

O resultado positivo apresentado por [Fradkin 2008] foi uma das motivações para a adoção de uma abordagem semelhante proposta neste trabalho. Contudo, é necessário ressaltar que a técnica proposta pelo autor utiliza toda a base de dados como entrada única para o treinamento de um método capaz de lidar com múltiplas classes. Assim, com um

método de classificação linear (no caso, o SVM Linear), múltiplos classificadores são gerados e combinados através da técnica *one-vs-all*. No final, é obtido um classificador para cada classe da base de dados [Zhouyu Fu et al. 2010].

Apesar da abordagem proposta neste trabalho também empregar a estratégia de divisão-e-conquista através do agrupamento dentro das classes, os grupos formados não são utilizados para rotular as amostras. Ao invés disso, a etapa posterior ao agrupamento realiza a interseção entre os grupos das partições formadas, mantendo os rótulos originais e produzindo uma única partição final com diversos subconjuntos da base completa. O treinamento é realizado em cada subconjunto gerando classificadores lineares especializados. A abordagem proposta ainda produz vários classificadores no final, entretanto, como a quantidade de amostras em cada subconjunto é substancialmente reduzida, a etapa de treinamento tende a ser muito mais rápida [Fradkin 2008, Boley and Cao 2004]. Além disso, tanto a etapa de treinamento quanto de interseção podem ser paralelizadas para assegurar maior eficiência.

A proposta deste trabalho, portanto, incorpora as principais vantagens presentes nos trabalhos correlatos existentes na literatura, uma vez que a estratégia de agrupamento dentro das classes é utilizada em conjunto com as estratégias de geração de classificadores especializados e seleção de modelos [Kuncheva 2000, Britto et al. 2014]. Com isso, espera-se que classificadores lineares treinados com diferentes visões localizadas das amostras possam, em conjunto, oferecer desempenho similar ao de um classificador não-linear, com a vantagem de ser muito mais eficiente em termos de complexidade computacional, além da possibilidade de paralelizar as etapas mais custosas.

3. Abordagem proposta

A abordagem proposta neste trabalho é dividida em quatro etapas: (i) particionamento por classe, (ii) interseção dos grupos, (iii) treinamento local e (iv) classificação. Cada uma delas é detalhada a seguir.

3.1. Particionamento por classe

A primeira etapa consiste na separação das amostras de acordo com os seus rótulos e, em seguida, na aplicação de um algoritmo de agrupamento sobre os dados. O objetivo desse procedimento é tentar identificar estruturas particulares de cada classe, que não seriam encontradas caso o agrupamento fosse aplicado sobre a base toda. A Figura 2 ilustra o resultado da aplicação de um algoritmo de agrupamento por densidade nas duas classes da base de dados ilustrada na Figura 1(a).

O algoritmo de agrupamento escolhido para esta etapa foi o tradicional k -médias, por ser simples e normalmente apresentar bom desempenho, além de ser um dos mais indicados e utilizados na literatura [Arthur and Vassilvitskii 2007, Fradkin 2008]. Por não ser determinístico, o k -médias pode ser executado várias vezes e o resultado selecionado será aquele que apresentar o menor erro quadrático médio da distância das amostras em relação ao centroide mais próximo [Arthur and Vassilvitskii 2007]. O parâmetro k (quantidade de grupos) pode ser determinado pelo método do cotovelo, que consiste em variar o valor de k em um intervalo e em seguida, escolher o valor que minimiza significativamente a soma do erro quadrático médio, ou seja, o maior valor de k que adiciona no erro médio uma variância maior que um certo limiar.

Após a aplicação do algoritmo de agrupamento, é possível que existam grupos com um número muito pequeno de amostras, que precisam ser tratados para evitar o desbalanceamento na etapa de treinamento. Um limite mínimo de amostras por grupo foi

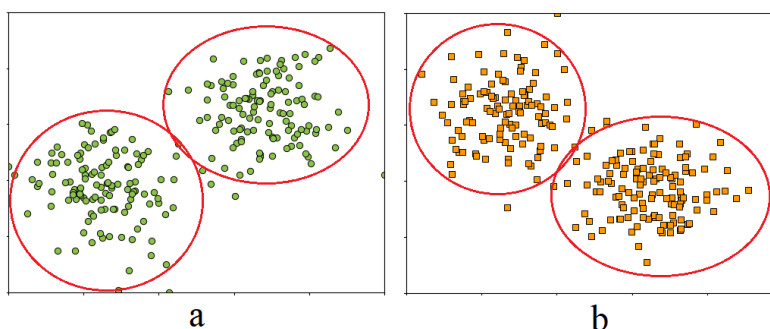


Figura 2. Particionamento por classe da base de dados ilustrada na Figura 1(a). Cada círculo representa um grupo de amostras encontrado pelo algoritmo de agrupamento baseado em densidade. Neste exemplo, foram encontrados dois grupos em cada uma das classes (partições).

definido como um parâmetro para evitar esse problema. Assim, os grupos considerados pequenos são unidos aos grupos mais próximos, de acordo com a distância euclidiana entre seus respectivos centroides.

3.2. Interseção dos grupos

Seja K o número de classes em uma determinada base de dados. Após a aplicação do algoritmo de agrupamento sobre as amostras de cada classe, são produzidas K partições $\mathcal{P} = \{p_1, \dots, p_K\}$. Uma partição p_i possui k grupos disjuntos $\{g_{i_1}, \dots, g_{i_k}\}$. A etapa de interseção tem o objetivo de identificar e unir grupos de diferentes classes cujas amostras estejam sobrepostas, para comporem o conjunto de treinamento de um mesmo classificador. Com isso, espera-se obter um classificador linear especializado na predição de amostras que estão localizadas em uma região de difícil separação (com sobreposição).

A interseção entre os grupos de partições de cada classe é feita em pares, comparando todos os grupos de uma partição com os grupos de outra. A união de dois grupos $g_{ij} \in p_i, j \in 1 \dots k_i$ com $g_{lm} \in p_l, m \in 1 \dots k_l$, sendo que $i \neq l$ e $p_i, p_l \in \mathcal{P}$, é realizada de acordo com um fator de interseção L que depende dos valores obtidos pela sobreposição S e proporção P entre cada par de grupos g_{ij} e g_{lm} .

A *sobreposição* (S) entre dois grupos quaisquer g_i e g_l corresponde à razão entre a quantidade de amostras sobrepostas e o total de amostras $|g_i \cup g_j|$ (Equação 1). Para determinar a quantidade de amostras sobrepostas, utiliza-se o raio de maior comprimento de cada grupo, isto é, a distância euclidiana entre o centroide e a amostra do grupo que está mais distante dele. Isto quer dizer que, cada uma das amostras de g_i que esteja dentro do raio de maior comprimento de g_l é contada como sobreposta, e vice-versa.

$$S_{g_i, g_l} = \frac{\text{n}^\circ \text{ de amostras sobrepostas}}{|g_i \cup g_j|} \quad (1)$$

A *proporção* (P) entre dois grupos quaisquer corresponde à razão entre a densidade do grupo menor (com menos amostras) pelo grupo maior (com mais amostras) (Equação 2). A densidade de um grupo se refere ao número de amostras do grupo, normalizado entre 0 e 1, de modo que o grupo com mais amostras possua densidade igual a 1. Com isso, se os grupos comparados forem proporcionais, o resultado de P será próximo de 1, caso contrário tenderá a 0.

$$P_{g_i, g_l} = \frac{\text{densidade de } |g_i|}{\text{densidade de } |g_l|}, \text{ sendo } |g_i| \leq |g_l| \quad (2)$$

O fator de interseção L_{g_i, g_l} entre dois grupos g_i e g_l é calculado pela divisão do valor da sobreposição (S_{g_i, g_l}) pela proporção (P_{g_i, g_l}), conforme a Equação 3.

$$L_{g_i, g_l} = \frac{S_{g_i, g_l}}{P_{g_i, g_l}}, \text{ com } P_{g_i, g_l} \neq 0 \quad (3)$$

A Tabela 1 analisa todas as possibilidades de resultados para o fator de interseção e apresenta os casos considerados para união de g_i e g_l , sendo que $0 \leq S_{g_i, g_l} \leq 1$ e $0 < P_{g_i, g_l} \leq 1$. Valores de S_{g_i, g_l} tendendo a 1, indicam que há muita sobreposição entre as amostras, e valores tendendo a 0 indicam pouca sobreposição. Analogamente, valores P_{g_i, g_l} tendendo a 1 indicam que os grupos são proporcionais e valores tendendo a 0, que os grupos são desproporcionais.

S_{g_i, g_l}	P_{g_i, g_l}	L_{g_i, g_l}	União
$S_{g_i, g_l} = 0$	$P_{g_i, g_l} > 0$	$L_{g_i, g_l} = 0$	Não
$S_{g_i, g_l} \rightarrow 0$	$P_{g_i, g_l} \rightarrow 1$	$L_{g_i, g_l} \rightarrow 0$	Não
$S_{g_i, g_l} \rightarrow 0$	$P_{g_i, g_l} \rightarrow 0$	$L_{g_i, g_l} \rightarrow 1$	Sim
$S_{g_i, g_l} \rightarrow 1$	$P_{g_i, g_l} \rightarrow 1$	$L_{g_i, g_l} \rightarrow 1$	Sim
$S_{g_i, g_l} \rightarrow 1$	$P_{g_i, g_l} \rightarrow 0$	$L_{g_i, g_l} \rightarrow +\infty$	Sim

Tabela 1. Possíveis valores para o limiar de interseção. A última coluna indica os casos em que a união é aplicada.

É possível observar que a união é aplicada sempre que há sobreposição de muitas amostras, ou ainda, quando há alguma sobreposição e o tamanho dos grupos é desproporcional (um deles muito maior que o outro). Em todos os casos, o critério para união é definido por meio de um parâmetro de entrada (γ), que define um limiar de decisão para determinar se a união deverá ou não ser aplicada, $L_{g_i, g_l} \geq \gamma$.

A Figura 3 ilustra um exemplo no qual dois grupos foram formados após a etapa de interseção aplicada entre os grupos das duas partições ilustradas na Figura 2. Nota-se que, como neste caso os grupos são proporcionais, a interseção foi definida pela sobreposição e, no final, dois grupos distintos compuseram a partição resultante.

3.3. Treinamento

Na etapa de treinamento, um método de classificação é utilizado para gerar um classificador para cada um dos grupos da partição final resultante da etapa de interseção. O objetivo é produzir um classificador especializado nas amostras de cada grupo, e assim, poder oferecer melhor desempenho na predição das classes.

É possível que alguns dos grupos encontrados sejam constituídos por amostras de uma única classe. Neste caso, o próprio grupo torna-se em um classificador óbvio. Assim, se uma nova amostra de teste pertencer a algum desses grupos, ela é classificada como sendo da mesma classe de todas as amostras do grupo.

Caso existam grupos claramente desbalanceados, um algoritmo de balanceamento é aplicado para adicionar ou remover amostras e assim, balancear o grupo de modo que

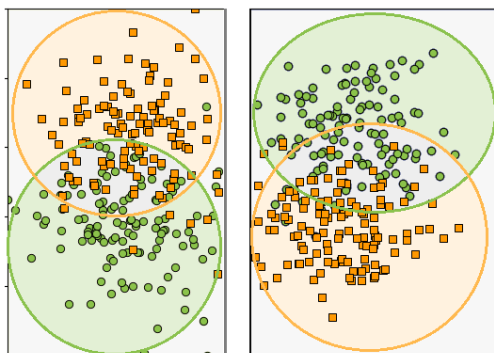


Figura 3. Resultado da interseção aplicada nos grupos das partições ilustradas na Figura 2. A partição final é formada por dois grandes grupos representados pelos retângulos.

o treinamento não seja impactado negativamente. Para realizar esse balanceamento, foi utilizada a biblioteca *imbalanced-learn*¹ [Lemaitre et al. 2016]. Contudo, caso existam grupos extremamente desbalanceados, de modo que o algoritmo de balanceamento não consiga tratá-los, nenhum classificador é treinado e a classe majoritária do grupo é atribuída para as novas amostras de teste pertencentes a ele.

3.4. Classificação

A fase de treinamento faz com que cada grupo da partição final seja associado a um mecanismo de classificação das amostras de teste que porventura sejam atribuídas a ele. Esse mecanismo pode ser a atribuição direta da classe do grupo ou o emprego de um classificador especializado para o grupo. Para classificar uma nova amostra, primeiramente, é necessário identificar a qual grupo ela pertence. Isto é feito a partir do cálculo da distância euclidiana da amostra em relação aos centroides de cada grupo. Ao encontrar o centroide mais próximo, o mecanismo de classificação associado àquele grupo é utilizado para classificar a nova amostra.

A Figura 4 ilustra os mecanismos de classificação definidos para a partição final, resultante do treinamento aplicado na partição apresentada na Figura 3. É possível notar que existem dois grupos, indicados pelos seus respectivos centroides (indicados por ‘×’). Para cada um deles, um classificador linear foi treinado, de modo que cada nova amostra de teste é associada ao centroide mais próximo e, conseqüentemente, a um dos classificadores especializados.

4. Experimentos e resultados

Os experimentos foram conduzidos com dois objetivos distintos:

1. Comparar o desempenho dos métodos lineares de classificação com e sem o uso da abordagem proposta neste trabalho, quando aplicados em bases de dados compostas por amostras não separáveis linearmente; e
2. Verificar se a abordagem proposta pode conduzir à combinação de classificadores lineares que ofereça desempenho de predição similar a de um classificador não-linear.

A seguir, são oferecidos detalhes sobre a metodologia experimental e os resultados obtidos.

¹A implementação da biblioteca *imbalanced-learn* está disponível em: <https://goo.gl/nfjZMX>. Acesso em 29/05/2017.

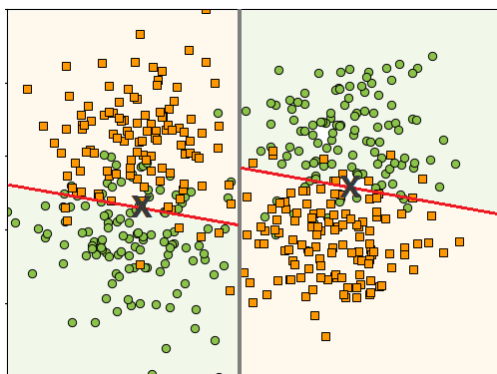


Figura 4. Modelo de classificação resultante da aplicação da abordagem proposta. A linha vertical representa a divisão entre os grupos da partição (indicados pelos centroides ‘x’) e as linhas vermelhas representam os classificadores lineares treinados para cada grupo. Cada amostra nova de teste é classificada a partir do preditor correspondente ao grupo a que ela pertencer.

4.1. Bases de dados

Para assegurar uma avaliação robusta da abordagem proposta, foram realizados testes com 7 bases de dados artificiais, heterogêneas, multi-classe, com diferentes estruturas de grupos e com dados não separáveis linearmente². A Figura 5 ilustra cada uma dessas bases, sendo que cada cor representa uma classe distinta.

4.2. Metodologia experimental

Os métodos lineares de classificação utilizados foram: *Logistic Regression* (LR) e *Linear SVM* (LSVM). Além desses, o método probabilístico *Bernoulli Naïve Bayes* (BNB) também foi avaliado. Os resultados obtidos com a abordagem proposta foram comparados com o desempenho do método *Radial SVM* (RSVM).

Para a implementação dos métodos de classificação, foi utilizada a biblioteca *scikit-learn*³ [Pedregosa et al. 2011], sendo que os meta-parâmetros foram ajustados através de *grid-search* e validação cruzada com 5 fold: o método BNB foi treinado variando o limiar de suavização no intervalo entre 10^{-7} e 10^7 e o limiar de binarização no intervalo entre 10^{-7} e 10^7 ; os métodos LR e LSVM foram avaliados variando o parâmetro de regularização (C) no intervalo entre 10^{-3} e 10^3 ; já o método RSVM foi avaliado variando os parâmetros C e γ no intervalo de 10^{-3} a 10^3 .

Para a etapa de agrupamento, o k -médias foi executado 50 vezes, variando o número de grupos no intervalo entre 1 e 20. O limiar de variância empregado no método do cotovelo foi de 10%. Cada grupo resultante possui um número mínimo de 20 amostras, valor definido empiricamente, a fim de evitar a insuficiência de amostras durante o procedimento de validação cruzada no treinamento dos métodos de classificação. O parâmetro de entrada do método referente ao limiar de interseção foi empiricamente definido como $\gamma = 0,5$.

A base de dados foi dividida em 5 fold, de modo que, o experimento fosse repetido 5 vezes e em cada repetição do experimento, 80% fosse utilizado para o treinamento e 20%

²Todas as bases de dados estão disponíveis em: <https://bitbucket.org/nahimsouza/mestrado>. Acesso em 29/05/2017.

³A biblioteca *scikit-learn* está disponível em: <http://scikit-learn.org/stable/>. Acesso em 29/05/2017.

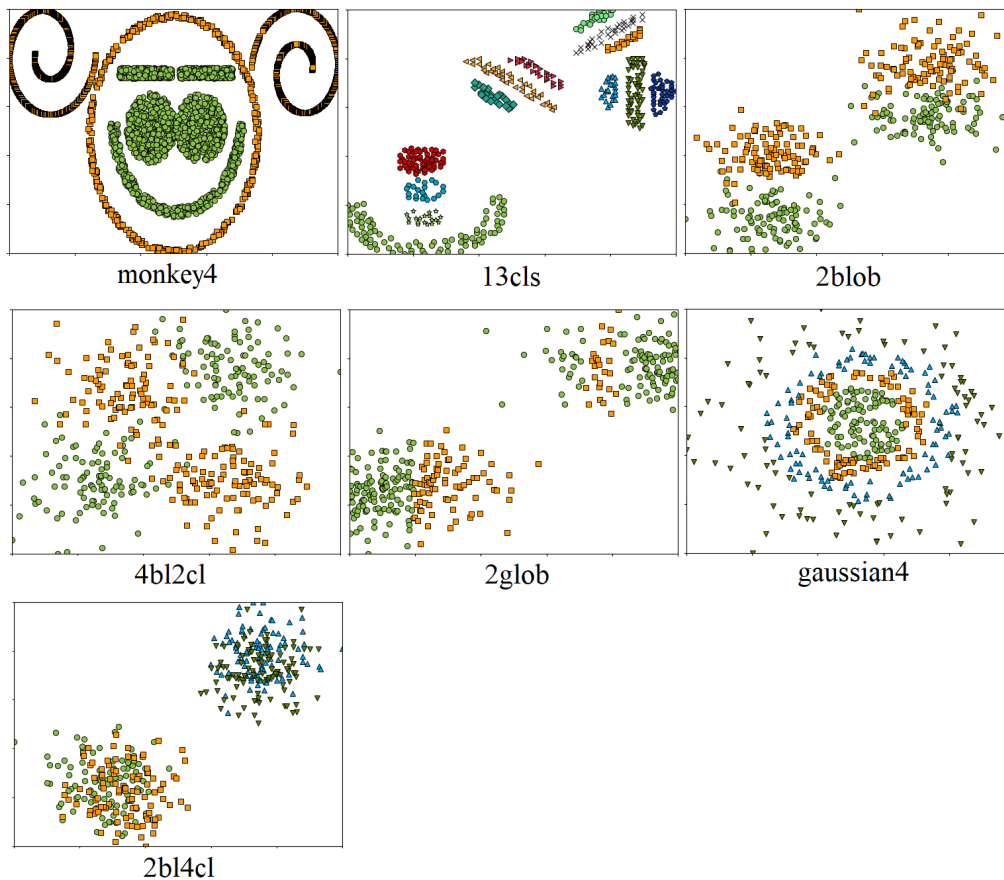


Figura 5. Bases de dados utilizadas nos experimentos.

para testes (Figura 6). O objetivo de abordagem foi garantir que as amostras utilizadas durante o treinamento e o teste dos classificadores fossem as mesmas, para permitir uma melhor comparação entre os resultados. É importante notar que a validação cruzada é realizada em dois momentos: (1) externamente, para avaliar o desempenho dos métodos e (2) internamente, para o treinamento dos classificadores.

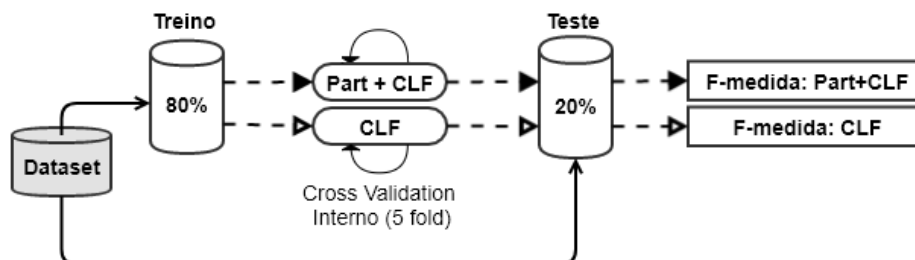


Figura 6. Metodologia experimental. *CLF* corresponde ao método de classificação tradicional. *Part+CLF* representa a abordagem proposta neste trabalho.

4.3. Análise dos resultados

A Tabela 2 apresenta a comparação de desempenho obtido pelos métodos de classificação com e sem o uso da abordagem de particionamento proposta neste trabalho. Para essa

avaliação a métrica utilizada foi a F-medida, calculada a partir da média obtida em cada uma das partições de teste da validação cruzada (Figura 6). As primeiras três linhas indicam os resultados da macro F-medida média obtida pelos métodos tracionais. As três linhas seguintes apresentam a macro F-medida média obtida pelos mesmos métodos utilizando a abordagem de particionamento proposta ('Part'). A última linha da tabela apresenta a macro F-medida média obtida pelo método SVM com *kernel* radial.

	monkey4	13cls	2blob	4bl2cl	2glob	gaussian4	2bl4cl
BNB	0,58	0,09	0,50	0,41	0,41	0,18	0,24
LR	0,61	0,45	0,88	0,54	0,60	0,23	0,61
LSVM	0,66	0,88	0,88	0,62	0,41	0,31	0,57
Part+BNB	0,96	0,52	0,65	0,88	0,51	0,35	0,44
Part+LR	1,00	0,96	0,91	0,90	0,86	0,62	0,62
Part+LSVM	1,00	0,96	0,93	0,89	0,85	0,60	0,59
RSVM	1,00	0,97	0,90	0,91	0,92	0,67	0,59

Tabela 2. Macro F-medida média obtida pelos métodos de classificação avaliados. Os melhores resultados para cada base de dados estão destacados em negrito.

É possível observar que, comparando o desempenho dos métodos avaliados, a combinação dos classificadores por grupo após o particionamento por classe conduziu a uma melhoria significativa no desempenho dos métodos em todas as bases de dados utilizadas. Além disso, na maioria dos testes executados, a abordagem proposta proporcionou que os métodos oferecessem resultados próximos ou até mesmo superior ao do método não-linear (RSVM).

Para analisar o quanto a abordagem proposta favoreceu o desempenho dos métodos avaliados, a Tabela 3 apresenta o ganho em %, dado pela diferença percentual entre a macro F-medida média obtida pelos métodos com e sem o emprego da abordagem de particionamento.

	monkey4	13cls	2blob	4bl2cl	2glob	gaussian4	2bl4cl
Part+BNB - BNB	65,5	477,8	30,0	114,6	24,4	94,4	83,3
Part+LR - LR	63,9	113,3	3,4	66,7	43,3	169,6	1,6
Part+LSVM - LSVM	51,5	9,1	5,7	43,5	107,3	93,6	3,5

Tabela 3. Diferença de desempenho (em %) entre a macro F-medida média obtida pelos métodos de classificação avaliados com e sem o emprego da abordagem de particionamento proposta.

É evidente que os três métodos de classificação avaliados obtiveram ganhos consideráveis ao serem combinados com o particionamento por classe. Em vários casos, a capacidade de predição chegou a dobrar. Outro destaque é o desempenho do método BNB, que embora não tenha alcançado os melhores resultados, a aplicação do particionamento por classe fez com que a macro F-medida média aumentasse significativamente em todas as bases de dados.

Para garantir que os resultados não foram obtidos ao acaso, foi realizada uma análise estatística seguindo cuidadosamente os passos descritos por [Demšar 2006]. Primeiramente, foi efetuado um *teste de hipótese* que avalia a probabilidade de se cometer

um erro tipo I (rejeição da hipótese nula quando ela é verdadeira). A hipótese nula, neste caso, assume que o aumento no desempenho dos métodos foi obtido ao acaso e não devido à aplicação da abordagem proposta.

Para realizar o teste de hipótese, primeiramente, é necessário estabelecer um nível de significância (α), que corresponde à probabilidade de se cometer um erro tipo I. Em seguida, a estatística do teste e o p -valor são calculados. O p -valor representa a probabilidade de significância e, caso o p -valor seja menor que o valor de α , o resultado é considerado como significativo e a hipótese nula é rejeitada.

Neste trabalho, foram utilizados o Teste-T e o Teste de Friedman. O Teste-T foi utilizado para comparar a variação do desempenho aos pares, usando os resultados com e sem a aplicação do particionamento por classe em cada um dos métodos de classificação avaliados. O Teste de Friedman, por sua vez, foi utilizado para avaliar a variabilidade entre os resultados obtidos.

Para ambos os testes, o nível de significância estabelecido foi de 5%, isto é, $\alpha = 0,05$. O aumento médio no desempenho reportado na Tabela 3 foi de 0,25. Tanto o Teste-T quanto o Teste de Friedman indicam que esse aumento no desempenho foi estatisticamente significativo, com p -valor $< 0,05$, e portanto, a hipótese nula pode ser seguramente rejeitada. Assim, de acordo com a análise realizada, há 95% de chance de se obter um aumento médio de 0,25 no desempenho (macro F-medida média), a partir da aplicação da abordagem de agrupamento proposta neste trabalho.

5. Conclusão

Neste trabalho, foi proposta uma nova abordagem baseada na estratégia de divisão-e-conquista para melhorar o desempenho de métodos de classificação lineares. A partir dos resultados, verificou-se que a aplicação da proposta conseguiu aumentar significativamente o poder preditivo desses métodos de classificação, mesmo em bases de dados multi-classe compostas por amostras não separáveis linearmente.

Os resultados experimentais evidenciam que o emprego da abordagem proposta permitiu que os métodos avaliados atingissem resultados de predição similares ou até mesmo superiores aos do SVM radial nas bases de dados investigadas. A melhoria no desempenho foi muitas vezes significativo e comprovado através da análise estatística dos resultados. Além dos benefícios proporcionados à capacidade de predição, vale ressaltar que a abordagem proposta oferece a possibilidade de paralelização das etapas de treinamento para cada grupo encontrado na partição final. Com isso, aliado à menor quantidade de amostras por grupo, o treinamento de cada método linear além de ser acelerado, pode ser massivamente paralelizado. Tais características são altamente desejáveis em aplicações reais, nas quais muitas vezes os métodos de predição não-lineares são computacionalmente custosos.

Atualmente, estão sendo realizados testes com bases de dados reais e de grande porte, além de estudos relacionados à paralelização das etapas de interseção e treinamento. Para trabalhos futuros, outros métodos de classificação devem ser avaliados com o objetivo de aprofundar a comparação com o método proposto.

Referências

Almeida, L. M. e Galvão, P. S. (2016). Ensembles with clustering-and-selection model using evolutionary algorithms. In *Proceedings of the 5th Brazilian Conference on Intelligent Systems (BRACIS'16)*, Recife, Brazil.

- Arthur, D. e Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM symposium on Discrete algorithms (SODA'07)*, volume 8, pages 1027–1035, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- Boley, D. e Cao, D. (2004). Training Support Vector Machine using Adaptive Clustering. In *Proceedings of the 4th SIAM International Conference on Data Mining (SDM'04)*, pages 126–137, Philadelphia, PA. Society for Industrial and Applied Mathematics.
- Britto, A. S., Sabourin, R., e Oliveira, L. E. (2014). Dynamic selection of classifiers—A comprehensive review. *Pattern Recognition*, 47(11):3665–3680.
- Chih-Wei Hsu e Chih-Jen Lin (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.
- Coletta, L. F. S., Silva, N. F. F. D., Hruschka, E. R., e Hruschka, E. R. (2014). Combining Classification and Clustering for Tweet Sentiment Analysis. In *Proceedings of the Brazilian Conference on Intelligent Systems (BRACIS'14)*, pages 210–215. IEEE.
- Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30.
- Fradkin, D. (2008). Clustering Inside Classes Improves Performance of Linear Classifiers. In *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI08)*, volume 2, pages 439–442. IEEE.
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., e Hinton, G. E. (1991). Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79–87.
- Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '06*, pages 217–226, Philadelphia, PA, USA. ACM Press.
- Kuncheva, L. (2000). Clustering-and-selection model for classifier combination. In *Proceedings of the 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies (KES'00)*, volume 1, pages 185–188, Brighton, UK. IEEE.
- Kyriakopoulou, A. e Kalamboukis, T. (2008). Combining Clustering with Classification for Spam Detection in Social Bookmarking Systems. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD'08)*, pages 47–54, Antwerp, Belgium.
- Lemaitre, G., Nogueira, F., e Aridas, C. K. (2016). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, 18:1–5.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., e Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Ruta, D. e Gabrys, B. (2005). Classifier selection for majority voting. *Information Fusion*, 6(1):63–81.
- Zhouyu Fu, Robles-Kelly, A., e Jun Zhou (2010). Mixing Linear SVMs for Nonlinear Classification. *Journal of IEEE Transactions on Neural Networks*, 21(12):1963–1975.

Online Modified Differential Evolution Method

Static Stability Control of Two-Wheeled Inverted Pendulum Vehicle Case Study

Bruno L. Pereira, Gustavo Diniz, Leonardo Sanches, José J.P.Z.S. Tavares

Faculdade de Engenharia Mecânica
Universidade Federal de Uberlândia (UFU) – Uberlândia, MG – Brazil

{lsanches,jean.tavares}@ufu.br, brunolp.meca@gmail.com,
gustavodiniz@ymail.com

***Abstract.** This paper presents a method called Online Modified Differential Evolution or OnMoDE, which is a real time updater of intelligent controllers parameters based on the differential evolution optimization technique. OnMoDE is applied as a case study in the static stability of a two-wheeled inverted pendulum vehicle from a neuro-fuzzy controller. Both, numerical and experimental tests, verify the effectiveness of the proposed method by the progressive reduction of the process cost function as time progresses.*

1. Introduction

The intelligent control techniques are based on the observation of how living beings, and in particular humans, behave facing and solving different problems. These techniques are generally versatile and adjusted for controlling different dynamical systems, even under any mechanical properties variation. Indeed, these variations can happen, as for example, due to aging or failure of mechanical components. In this sense, the control parameters need to be accessed and adapted during real time or in offline operation.

Among the techniques of intelligent control, the neuro-fuzzy system [Jang 1992] is quite applicable due to the adaptability of the neural network [Haykin 2008] combined with the robustness of the fuzzy set theory [Zadeh 1965]. It has been commonly used in both, the modeling [Babuska and Verbruggen 2013, Aquil et al. 2016] and the control of dynamical systems [Jang 1992, Godjevac and Steele 1993, Jang and Sun 1995, Saifizul et al. 2006, Chen et al. 2008, Rusu and Petriu 2008, Wannige and Sonnadara 2009, Han et al. 2013, Nejadfard et al. 2013, Petchinathan et al. 2014].

In the literature, the neuro-fuzzy control design is developed based on numerical computation for dynamical systems in which the model of the plant is known [Jang 1992, Saifizul et al. 2006, Chen et al. 2008, Han et al. 2013]. Other investigations propose the controlling parameters estimation from the experimental data through definition of the plant pseudo-inverse function [Jang and Sun 1995, Petchinathan et al. 2014]. Moreover, analytical method of backpropagation error, defined by the difference between the system's output and the setpoint, is applied for real time updating of the controlling parameters [Godjevac and Steele 1993, Rusu and Petriu 2008, Wannige and Sonnadara 2009, Nejadfard et al. 2013].

The present work proposes the development of a non-analytical method to be applied in real time adjustment of neuro-fuzzy controllers' parameters, named OnMoDE

(Online Modified Differential Evolution). The neuro-fuzzy control technique is combined with the differential evolution algorithm due to its relative simplicity and effectiveness at reaching the optimum global value with respect to others evolutionary methods [Storn and Price 1996, Chen et al. 2008, Han et al. 2013].

Later, for the validation of the proposed OnMoDE technique, the static stability of a two-wheeled inverted pendulum vehicle is treated in this paper. Due to the nonlinear dynamic behavior of the above mentioned vehicle, it has been currently used in the survey of benchmarking regarding the efficiency of controllers applied to nonlinear systems [Jang 1992, Saifizul et al. 2006, Bonafilia et al. 2014, Pereira et al. 2016]. The developed control technique is tested by numerical simulations that take into account the dynamic model of the system, and then, experimental tests are performed with the two-wheeled vehicle. The results are then compared with those obtained from a PID controller developed by root locus method to verify the technique efficiency in nonlinear systems control.

This paper begins, at section II, with the neuro-fuzzy system presentation, followed by a revision of the differential evolution algorithm. At section III, a description of the proposed control technique is presented, followed by the mechanical modeling of a two-wheeled vehicle. Then, there is the presentation of the work methodology, the results and discussions, and finally the conclusions.

2. Neuro-Fuzzy System

The adaptive neuro-fuzzy system, also known as ANFIS, arose from the merging of artificial neural networks ANN with the Fuzzy Inference System FIS. In this way, the ANFIS inherits the FIS structure, that is, it contains the fuzzification process, a rule base of type If ... Then, a fuzzy inference machine and the defuzzification; but with the same structure representation, learning capacity and adaptation of ANN through the adjustment of the parameters related to the fuzzy system.

Figure 1 sketches the structure representation of a neuro-fuzzy system that uses the fuzzy inference method of Takagi-Sugeno [Chen et al. 2008], where the variables $I_i(k), i = 1, \dots, n$, are the process inputs in iteration k ; $MF_{ij}, i = 1, \dots, n, j = 1, \dots, m$, are the membership functions associated with the input variables; $R_L, L = 1, \dots, R$, are the rules belonging to the ANFIS rule base and $f_L, L = 1, \dots, R$, are the functions that relate the input variables to the output variable $O(k)$.

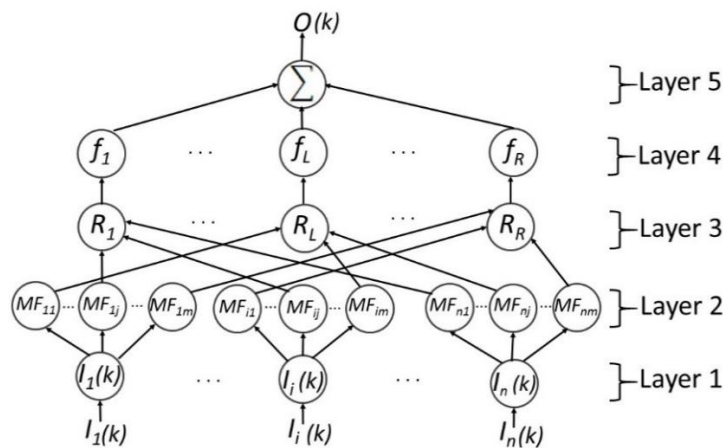


Figure 1. Neuro-fuzzy system representation.

Considering that the fuzzy inference method used is the first order Takagi-Sugeno, the functions f_L are then given by:

$$f_L = w_{0L} + w_{1L}I_1 + \dots + w_{iL}I_i + \dots + w_{nL}I_n \quad (1)$$

The parameters w_{iL} are weights whose values are obtained from some optimization method that minimizes the cost function Cf related to the process.

3. Differential Evolution

The differential evolution, algorithm developed by Storn and Price (1996), consists in a method for minimizing functions, which can be even nonlinear and non-differentiable. The algorithm is capable for arriving at the global optimum value even more easily and quickly than others classical and heuristic methods. According to [Storn and Price 1996], this method requires few control variables, it is robust, easy to implement and presents good adaptability to parallel computing.

The differential evolution algorithm presents the initialization, mutation, crossover, and selection steps [Han et al. 2013].

During the algorithm initialization, the differential evolution parameters are defined, and the search limits for each variable are determined, too. The vectors $X_{i,0}$, $i = 1, \dots, n_i$, where n_i is the number of vectors belonging to the set S of the possible solutions to the optimization problem, each represent a possible solution, and whose elements are normalized values of the process variables. The elements are initialized following uniform distribution defined within the interval $[0,1]$, and then the fitness of the vectors with respect to the function to be minimized, named fitness function fit , is calculated.

In the mutation step, the operation indicated in Eq. (2) is done, where $V_{i,k}$ is the resulting vector from the mutation operator, F is the perturbation factor of $X_{i,k}$, usually having a value contained in the interval $[0,2]$, and $X_{j,k}$ and $X_{l,k}$ are other randomly chosen vectors belonging to the set S . This procedure is performed n_i times in each iteration k .

$$V_{i,k} = X_{i,k} + F(X_{j,k} - X_{l,k}) \quad (2)$$

In the crossover step, it's merged the information present in vectors $V_{i,k}$ and $X_{i,k}$, guaranteeing greater diversity in the population. For each element e of vector $V_{i,k}$, a generator of values uniformly distributed within the interval $[0,1]$, $rand(0,1)$, is used and it's compared with the crossover rate CR value, also defined within the range $[0,1]$. The decision-making of this operator is specified in Eq. (3), where $C_{i,k}$ is the candidate vector to participate in the set S in $k + 1$ iteration and $randnum(1, n_e)$, where n_e is the number of elements of the vectors, is a generator of integers uniformly distributed within the interval $[1, n_e]$. This procedure is repeated for all vectors in the set S .

$$C_{e,i,k} = \begin{cases} V_{e,i,k}, & \text{if } rand(0,1) \leq CR \text{ or } e = randnum(1, n_e) \\ X_{e,i,k}, & \text{if } rand(0,1) > CR \text{ and } e \neq randnum(1, n_e) \end{cases} \quad (3)$$

Selection is the last standard operator of differential evolution and is based on the analysis of vectors $C_{i,k}$ generated at the crossover step. Equation (4) describes the selection procedure performed also for all vectors of the set S . Finishing this step, the next iteration $k + 1$ starts and the whole procedure is repeated. The algorithm will conclude the optimization process when some predetermined stop condition is met.

$$X_{i,k+1} = \begin{cases} X_{i,k}, & \text{if } fit(X_{i,k}) \geq fit(C_{i,k}) \\ C_{i,k}, & \text{another case} \end{cases} \quad (4)$$

4. Online Modified Differential Evolution Method

This section deals with the proposed method for real time update of the neuro-fuzzy control parameters based on the differential evolution optimization technique, method applied in nonlinear systems control.

The developed method is split into two stages, the first one deals with the initial generation of the vectors, and the second, the intelligent controller optimization process, which lasts throughout the plant operation after the vectors generation.

Figure 2 shows a flowchart representation of the Stage 1 algorithm. Each step of this algorithm is detailed in the following.

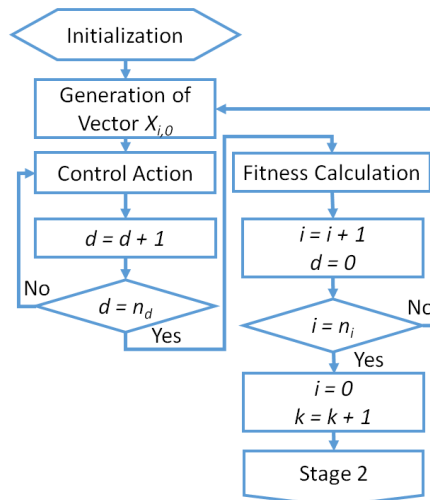


Figure 2. Neuro-fuzzy system OnMoDE stage 1 flowchart.

- **Initialization:** In this step, the number of vectors n_i of set S and the number of data n_d used to calculate the fitness of each vector are defined. It is also defined the initial value of the critical mutation $prob_{cmut}$, and finally, the variables i, d and k are initialized with null values;
- **Generation of Vector $X_{i,0}$:** The generation of each element of the vectors follows a uniform distribution defined between the interval 0 and 1, $rand(0,1)$, where the limit values 0 and 1 are related to the search interval limits of each variable related to the optimization problem;
- **Control Action:** This step is related with the determination of the control action applied in real time in the controlled system based on the parameters obtained from the vector $X_{i,k}$ (optimization process). Moreover, the experimental data are collected for performance evaluation of the controller;
- **Fitness Calculation of Vector $X_{i,0}$:** The fitness estimation of the vector $X_{i,0}$ with respect to the fitness function fit is computed. For this purpose, the experimental data are used.

Concerning the second stage of the OnMoDE algorithm, its flowchart representation is sketched in Figure 3:

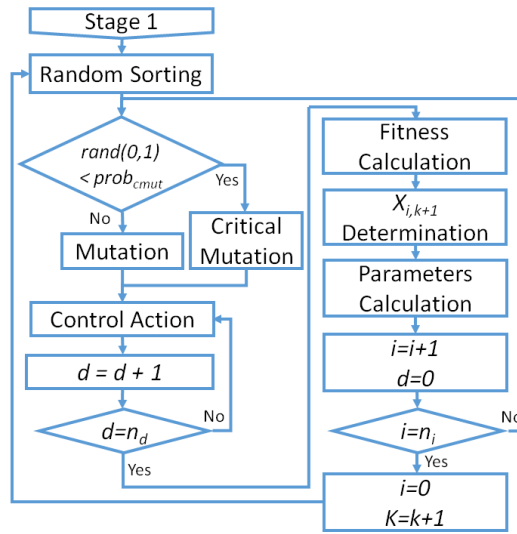


Figure 3. OnMoDE Stage 2 flowchart.

The main steps of this stage are described below:

- **Random Sorting:** In this step all possible mutations at k th iteration are performed based on the raffling of the vectors $X_{j,k}$ and $X_{l,k}$;
- **Mutation:** It consists in obtaining the candidate vector to participate in the iteration $k + 1$, $C_{i,k}$, from Eq. (5), where F also follows a uniform distribution defined between the interval 0 and 1:

$$C_{i,k} = X_{i,k} + F(X_{j,k} - X_{l,k}) \quad (5)$$
- **Critical Mutation:** The critical mutation promotes greater alteration in the vectors than the classical mutation operator does. This step can be developed in several ways, among which:

$$C_{i,k} = rand(0,1) \quad (6)$$

$$C_{i,k} = N(0.5, \sigma_{cmut}) \quad (7)$$

$$C_{i,k} = X_{i,k} + N(0, \sigma_{cmut}) \quad (8)$$
- **Control Action:** Identical to control action procedure described in stage 1 of the OnMoDE algorithm;
- **Fitness Calculation of Vector $C_{i,k}$:** Procedure analogous to the "Fitness Calculation of Vector $X_{i,0}$ " step in the Stage 1 of the OnMoDE algorithm;
- **Determination of $X_{i,k+1}$:** The determination of the vector $X_{i,k+1}$ is given from Eq. (4);
- **Parameters Calculation:** This step uses the information history of fitness function fit associated with vector $X_{i,k}$ to update the value of the parameters σ_{cmut} and $prob_{cmut}$. Examples of methods for updating these parameters are given below:

$$\sigma_{cmut} = \begin{cases} \sigma_1, & \text{if } \frac{fit(X_{i,k+1}) - fit(X_{i,k})}{fit(X_{i,k})} > ref_{cmut} \\ \sigma_2, & \text{if } \frac{fit(X_{i,k+1}) - fit(X_{i,k})}{fit(X_{i,k})} \leq ref_{cmut} \end{cases} \quad (9)$$

$$prob_{cmut} = \begin{cases} prob_1, & \text{if } \frac{fit(X_{i,k+1}) - fit(X_{i,k})}{fit(X_{i,k})} > ref_{cmut} \\ prob_2, & \text{if } \frac{fit(X_{i,k+1}) - fit(X_{i,k})}{fit(X_{i,k})} \leq ref_{cmut} \end{cases} \quad (10)$$

in which the values of σ_1 , σ_2 , prob_1 , prob_2 and ref_{cmut} are defined in the algorithm initialization step.

5. Two-Wheeled Inverted Pendulum Vehicle

For validation purpose of the new proposed control system algorithm, a two-wheeled inverted pendulum vehicle is used. This vehicle is modeled being x the vehicle position, \dot{x} its translational speed, θ is the system pitch angle, $\dot{\theta}$ the variation of the pitch angle in time and u is the supply voltage of the two DC motors. The dynamical system is represented by Eq. (11), accordingly to [Bonafilia et al. 2014].

$$\dot{\xi} = A\xi + Bu \quad (11)$$

with:

$$\xi = [x \ \dot{x} \ \theta \ \dot{\theta}]^T$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \alpha_M & \beta_M & \eta_M \\ 0 & 0 & 0 & 1 \\ 0 & \gamma_M & \delta_M & \vartheta_M \end{bmatrix}$$

$$B = [0 \ \alpha_M \varepsilon_M \ 0 \ \gamma_M \varepsilon_M]^T$$

The parameters $\alpha_M, \beta_M, \gamma_M, \delta_M, \varepsilon_M, \eta_M$ and ϑ_M are functions of the structure geometry, the properties of the materials of which the vehicle is composed and the properties of the vehicle's actuators.

It is important to note that the upright configuration of the two-wheeled inverted pendulum is an unstable equilibrium position [Saifizul et al. 2006]. The stabilization of the dynamical system at this position is possible through an application of a control technique.

6. Methodology

The effectiveness of the new proposed technique control, called OnMoDE algorithm, is verified on the static stability of the two-wheeled vehicle. The OnMoDE algorithm has the potential ability to adjust in real time the parameters of a neuro-fuzzy. Indeed, the self-adaptability of the control system are highly desired since the mechanical properties might constantly altered under normal operation.

In the following subsections the neuro-fuzzy controller, the simulation algorithm and the experimental procedure are described.

6.1. Neuro-Fuzzy Controller

The developed neuro-fuzzy controller is indicated as follows:

- **Input variables:** The normalized inputs variables at iteration k are: $u_{N,k}$, $\theta_{N,k}$ e $\Delta\theta_{N,k} = \theta_{N,k} - \theta_{N,k-1}$.
- **Input variables membership functions:** Each input variable is subdivided into 3 fuzzy subsets: low, regular and high, whose membership functions are Gaussian.
- **Output Variable:** Estimated voltage for motor power, normalized and at iteration $k+1$, $\hat{u}_{N,k+1}$;
- **Neuro-fuzzy system consequents:** Based on the first order Takagi-Sugeno method;

- **Rule base:** The neuro-fuzzy system rule base relates the input and output variables from a set of rules of type *IF...THEN*;
- **Neuro-fuzzy system training:** The weights of the neuro-fuzzy system consequents are given by the OnMoDE algorithm, which are determined with the minimization of the cost function Cf defined by Eq. (12):

$$Cf = \frac{\sum_{k=1}^{k_{max}} \lambda_1(u_k)^2 + \lambda_2(\theta_k - \theta_{ref})^2 + \lambda_3(\Delta\theta_k - \Delta\theta_{ref})^2}{k_{max}} \quad (12)$$

where θ_{ref} is the reference pitch angle, considered null, and $\Delta\theta_{ref}$ is the reference pitch angle variation, that is also considered null. The constants λ_1 , λ_2 and λ_3 are the weights of each term of Eq.(12) in the cost function. They needed to be designed in order to compromise the behavior of the response of the dynamical system and the control efforts.

6.2. Simulation Algorithm

Numerical simulations of the proposed control system applied for the static stability of the two-wheeled inverted pendulum are carried out before the experimental tests. For this purpose, the following conditions are considered:

- The vehicle operation is simulated for $T_t = 80$ s with a sampling time $T_s = 0.005$ s;
- It is assumed that the vehicle initial condition is only a pitch angle θ_0 ;
- The accelerometer is corrupted by zero mean white noise and standard deviation σ_{acc}
- Both θ_k and $\Delta\theta_k$ are corrupted respectively by zero mean white noise and standard deviation σ_θ and $\sigma_{\Delta\theta}$;
- Disturbances (e.g., impulse functions) are applied to the system pitch angle θ_k , with intensity i_θ and repeated n_{id} times;
- The mentioned above disturbances and noises are applied until the simulation time $T_p, T_p < T_t$. It's verified the controller ability to reject the system external disturbances.

The simulation considers the plant accordingly to the model presented in Eq. (11), and the neuro-fuzzy parameters update is performed by the OnMoDE method, which uses as fitness function *fit* the cost function Cf associated to the process, Eq. (12), and in order to prioritize the minimization of control effort and the maximization of the system energy efficiency, the values of λ_1 , λ_2 and λ_3 , respectively, are equal to 20, 2 and 0.5.

6.3. Experimental Procedure

The experimental evaluation of OnMoDE optimization method is done with a two-wheeled inverted pendulum, which presents an atmega328 as embedded microcontroller and a MPU6050 as inertial measurement unit. The experimental procedure seeks to reproduce some of the characteristics imposed in the simulation. They are:

- The two-wheeled vehicle begins its operation at rest, and with initial pitch angle θ_0 ;
- The sampling time adopted T_s is 0,005 s, and the vehicle operates during the total time interval $T_t = 80$ s;
- The accelerometer used is corrupted by zero-mean white noise and standard deviation σ_{acc} ;

During the system operation, the OnMoDE method is applied in real time, and again seeks to minimize the cost function Cf shown in Eq. (12), with the same values of λ_1 , λ_2 and λ_3 assumed in simulation.

7. Results and Discussions

The results obtained are presented into 3 subsections. The first one details the results obtained through numerical simulations while the second one shows experimental tests for the OnMoDE method performance evaluation. Finally, the third one, statistical tests are performed in order to infer on the controller ability to ensure the system static stability.

7.1. Simulation Results

Figures 4a and 4b show, respectively, the time history of the normalized control action and pitch angle, both calculated by simulating the two-wheeled vehicle operation under the action of OnMoDE neuro-fuzzy control. It is considered $n_i = 3$, $n_d = 3$, critical mutation performed from Eq. (6) and the parameter $probmut$ is computed through Eq. (10), where $prob_1 = 0.05$, $prob_2 = 0.01$ and $ref_{cmut} = 0.01$.

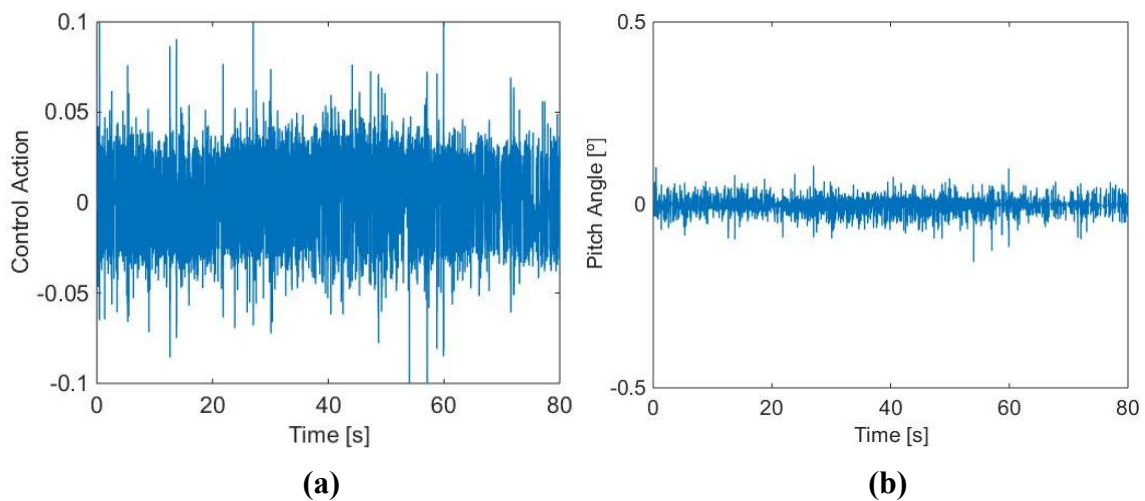


Figure 4. (a) Normalized control action and (b) pitch angle as a function of time obtained through simulation.

The cost function evolution in time is shown in Figure 5. One verifies a tendency to reduce the cost function amplitude as time is evolved. This highlights the performance of the proposed OnMoDE method, which is characterized by the online adjustment of the controller parameters for better performance.

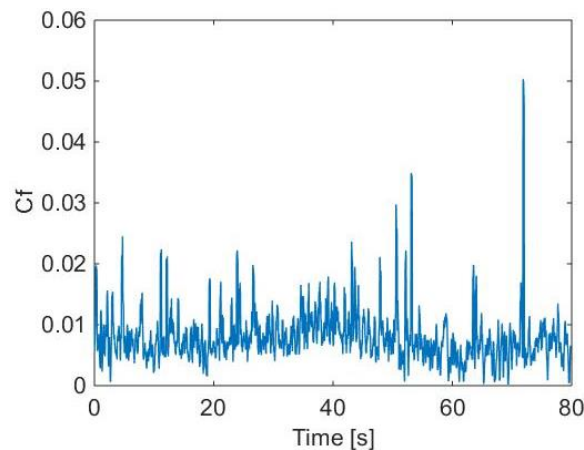


Figure 5. Evolution in time of the cost function value obtained through simulation.

7.2. Experimental Results

The OnMoDE method is applied in real time for controlling the two-wheeled vehicle. The experimental measured normalized control action and pitch angle are given in Figures 6a and 6b, respectively.

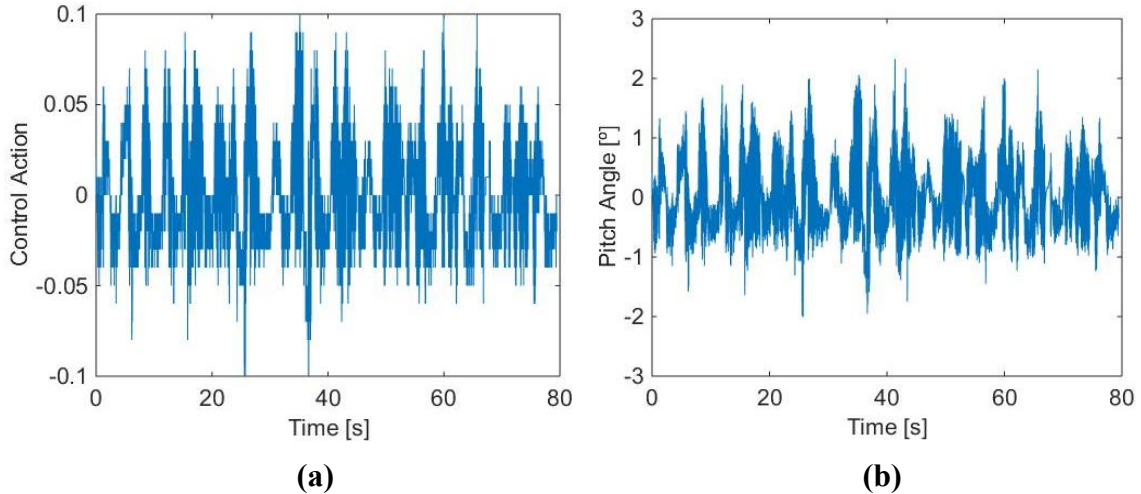


Figure 6. (a) Normalized control action and (b) pitch angle as a function of time.

The time evolution of the cost function is shown in Figure 7. Both, the cost function and the vehicle pitch angle are minimized as time progresses, which again is an indication of the self-tuning capability of neuro-fuzzy controller allied to OnMoDE method.

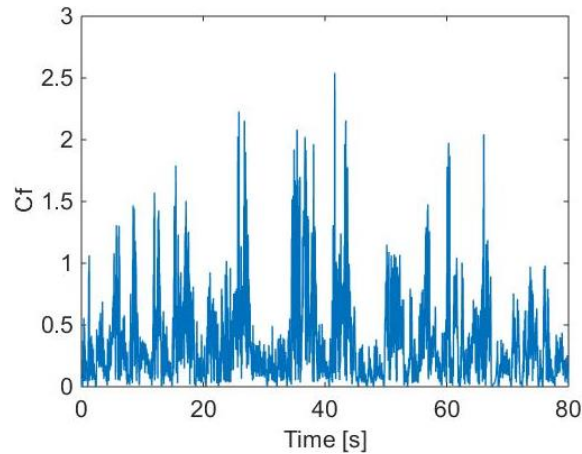


Figure 7. Evolution in time of the cost function value obtained experimentally.

In order to compare the performance of the controller allied to OnMoDE method, Figures 8a and 8b present respectively the normalized control action and pitch angle of the two-wheeled vehicle with the use of a PID controller designed from root locus method taking into account the dynamic model presented in Eq. (11).

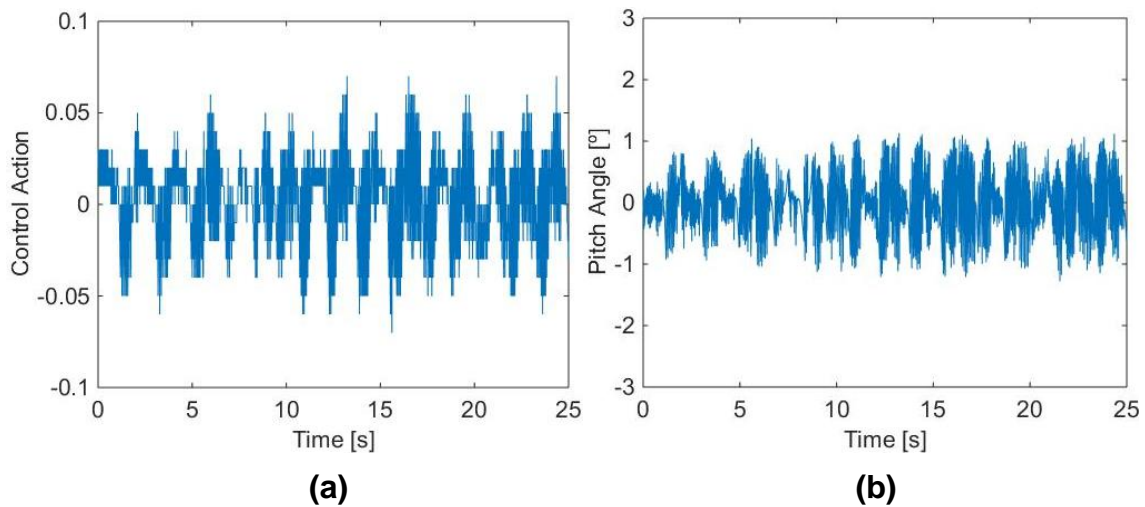


Figure 8. (a) Normalized control action and (b) pitch angle as a function of time using PID controller.

7.3. Statistical Analysis

Table 1 shows the sample means and standard deviations for the normalized control action \bar{u}_N and s_{u_N} , and for the pitch angle $\bar{\theta}$ and s_{θ_N} , both in the OnMoDE simulation and in the experiments using OnMoDE method and PID controller.

Table 1. Statistics Presentation

Statistic	OnMoDE Simulation	OnMoDE Experiment	OnMoDE Experiment (last 5 seconds)	PID controller
\bar{u}_N	0,00	0,00	0,00	0,00
s_{u_N}	$\pm 0,02$	$\pm 0,03$	$\pm 0,02$	$\pm 0,02$
$\bar{\theta}$	$0,00^\circ$	$0,03^\circ$	$-0,08^\circ$	$-0,01^\circ$
s_{θ}	$\pm 0,02^\circ$	$\pm 0,56^\circ$	$\pm 0,42^\circ$	$\pm 0,45^\circ$

From the data presented in Table 1, two statistical tests are performed in order to analyze the adaptive controller capacity to guarantee the two-wheeled vehicle static stability, which is verified by the control action and pitch angle minimization.

For the first statistical test, the controller's ability to minimize the control action of the vehicle, both in simulation and experimentally, is verified. Considering the t Student distribution, 2 hypotheses are elaborated:

- Null hypothesis: The normalized control action mean is zero;
- Alternative hypothesis: The normalized control action mean is nonzero.

In the second statistical test, the controller's ability to minimize the difference between the vehicle pitch angle and its respective zero reference value is analyzed, both in simulation and experimentally. Considering the t Student distribution, 2 hypotheses are also elaborated:

- Null hypothesis: The pitch angle mean is zero;
- Alternative hypothesis: The pitch angle mean is nonzero.

In both tests, the null hypothesis was not rejected with 95% confidence level, i.e., there is no indication that the adaptive controller is not able to statically stabilize the two-wheeled vehicle.

8. Conclusions

From the statistical tests performed, with 95% confidence, there is no evidence that the neuro-fuzzy controller tuned from the OnMoDE method is not able to guarantee the static stability of the two-wheeled inverted pendulum vehicle used as case study.

By comparing the values of s_θ obtained from the OnMoDE and PID methods, it is verified that the PID controller presents better results during practically the whole experiment. However, in the last 5 seconds the proposed approach has a lower pitch angle dispersion degree, which indicates the effectiveness of the method in adapting the consequent weights of the neuro-fuzzy controller in order to make it specific and efficient to the proposed application as time progresses.

The results and analyzes suggest that the OnMoDE method, due to its ability to impart adaptability to a control methodology, is a useful tool especially when it is desired to control complex systems, with difficult parameters obtainment or that they vary during the process, or when the plant dynamic model is unknown.

9. Further Work

The approach presented requires a new systematic evaluation of results to be developed. This systematic involves the area of machine learning and opens doors to new discoveries and functions with regard to self-adjusting control.

10. Acknowledgment

The authors acknowledge the Brazilian funding agencies CNPq, FAPEMIG and CAPES for the financial support for the development of this work.

References

- A. A. Saifizul et al., Intelligent Control for Self-erecting Inverted Pendulum Via Adaptive Neuro-fuzzy Inference System, American Journal of Applied Sciences, Malaysia, vol. 3, 2006, pp. 1795-1802.
- A. Nejadfard, M. J. Yazdanpanah, I. Hassanzadeh, Friction compensation of double inverted pendulum on a cart using locally linear neuro-fuzzy model, Neural Comput & Applic, London, vol. 22, 2013, pp. 337-347.
- B. Bonafilia et al., Self-balancing two-wheeled robot, 2014, pp. 1-11.
- B. L. Pereira et al., Two-Wheeled Vehicle Control Using Neural Network and Differential Evolution, Postgraduate Symposium in Mechanical Engineering of Federal University of Uberlandia (POSMEC), Brazil, 2016, pp. 1-7.
- C. H. Chen, C. J. Lin, C. T. Lin, Nonlinear System Control Using Adaptive Neural Fuzzy Networks Based on a Modified Differential Evolution, IEEE Transactions on Systems

- , Man, and Cybernetics — Part C: Applications and Reviews, Taiwan, vol. 39, 2008, pp. 459-473.
- C. T. Wannige, D. U. J. Sonnadara, Adaptive Neuro-Fuzzy Traffic Signal Control for Multiple Junctions, IEEE International Conference on Industrial and Information Systems (ICIIS), University of Colombo, vol. 4, 2009, pp. 262-267.
- G. Petchinathan et al., Local Linear Model Tree and Neurofuzzy System for Modelling and Control of an Experimental pH Neutralization Process, Brazilian Journal of Chemical Engineering, India, vol. 31, 2014, pp. 483-495.
- J. Godjevac, N. Steele, Neuro-Fuzzy Control of a Mobile Robot. Neurocomputing, Switzerland, v. 28, p. 127-143, jul. 1993.
- J. S. R. Jang, C. T. Sun, Neuro-Fuzzy Modeling and Control, Proceedings of the IEEE, UC Berkeley, vol. 83, 1995, pp. 378-406.
- J. S. R. Jang, Fuzzy Controller Design without Domain Experts, IEEE International Conference on Fuzzy Systems, Berkeley, 1992, pp. 289-296.
- L. A. Zadeh, Fuzzy Sets. Information and Control, Berkeley, vol. 8, 1965, pp. 338-353.
- M. Aquil et al., Analysis and Prediction of Flow from Local Source in a River Basin Using a Neuro-Fuzzy Modeling Tool, Journal of Environmental Management, Japan, vol. 85, 2016, pp. 215-223.
- M. F. Han, C. T. Lin, J. Y. Chang, Differential Evolution With Local Information for Neuro-Fuzzy Systems Optimisation. Knowledge-Based Systems, Taiwan, vol. 44, 2013, pp. 78-89.
- P. Rusu, E. M. Petriu, Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation, IEEE Transactions on Instrumentation and Measurement, Canada, vol. 52, 2008, p. 1335-1340.
- R. Babuska, H. Verbruggen, Neuro-fuzzy methods for nonlinear system identification. Annual Reviews in Control, Netherlands, vol. 27, 2013, pp. 73-85.
- R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization, Netherlands, vol. 11, 1996, pp. 341-359.
- S. Haykin, Redes neurais: Princípios e Prática, 2nd ed., Bookman, Canada, 2008, p. 899.

Novas Funcionalidade do SIMULA - Ambiente para Simulação em Sistemas Multiagentes Reativos

Bruno Pick¹, Rejane Frozza^{1,2}, Daniela Bagatini¹, Gilberto Dessbesell¹

¹Departamento de Computação – Universidade de Santa Cruz do Sul
Av. Independência, 2293 – Bairro Universitário – Santa Cruz do Sul – RS – Brazil

²Programa de Pós-Graduação em Sistemas e Processos Industriais – Universidade de Santa Cruz do Sul

brunopick@gmail.com, {frozza, bagatini}@unisc.br, gdbeto@gmail.com

Abstract. *SIMULA environment was designed for the development of reactive multiagent systems, being used to create applications through a graphical interface and with a real time visual simulation. The objective of this work was to incorporate SIMULA usability improvements and new functionalities, bringing interactive aspects to the simulation. The improvements implemented are the result of the ideas coming from the users of the tool and the analysis of other tools of the area. After the development the tool underwent a validation process, through the application of tests with a group of users, gathering the experience obtained with SIMULA.*

Resumo. *O ambiente SIMULA foi concebido para o desenvolvimento de sistemas multiagentes reativos, sendo utilizada para criar aplicações através de uma interface gráfica e com uma simulação visual em tempo real. O objetivo deste trabalho foi incorporar ao SIMULA melhorias de usabilidade e novas funcionalidades, trazendo aspectos interativos à simulação. As melhorias implementadas são resultado das ideias advindas dos usuários da ferramenta e da análise de outras ferramentas da área. Após o desenvolvimento a ferramenta passou por um processo de validação, através da aplicação de testes com um grupo de usuários, colhendo a experiência obtida com o SIMULA.*

1. Introdução

Um dos ambientes utilizados para simulação multiagente é o Ambiente para Simulação em Sistemas Multiagentes Reativos (SIMULA). Ele foi desenvolvido por [Frozza 1997], e permite a criação de aplicações em sistemas multiagentes reativos. Por meio de uma interface gráfica, o usuário define sua aplicação e define um modelo, ou seja, a representação de um problema real, visando atingir uma solução através da interação entre os agentes e o ambiente no qual estão inseridos.

Após quase 20 anos desde seu desenvolvimento, o SIMULA continua sendo utilizado no meio acadêmico em diversas universidades do Brasil como Universidade de São Paulo (USP), Universidade Federal do Rio Grande do Sul (UFRGS), Universidade de Rio Grande (FURG), UNISC, Universidade Federal de Santa Catarina (UFSC), entre outras, para o desenvolvimento de trabalhos na área. Sua simplicidade e facilidade de uso são as razões pelas quais o SIMULA é utilizado, pois o seu foco não é voltado à programação,

mas sim à modelagem e simulação de Sistemas Multiagentes (SMA), facilitando o aprendizado em simulações e SMA.

Um ponto relevante para se discutir no SIMULA é a questão de usabilidade, que não foi considerada em suas versões anteriores, mas é importante para o desempenho da ferramenta em relação a interação com seus usuários. O conceito de usabilidade [Krug 2006] consiste em fazer algo que funciona bem, de modo que mesmo uma pessoa não muito experiente consiga usá-lo e atinja a sua devida finalidade, levando em conta que durante esse processo não ocorra frustração.

O objetivo foi desenvolver novas funcionalidades ao SIMULA, tornando-o mais interativo nos aspectos referentes às simulações. Como problemática da pesquisa cita-se a necessidade de oferecer ferramentas com usabilidade para os usuários e mais interativas para configuração das simulações a serem realizadas. Desta forma, neste trabalho foram pesquisadas e desenvolvidas melhorias que agregaram alterações de usabilidade e novas funcionalidades ao SIMULA. Este artigo está organizado nas seguintes seções: A seção 2 apresenta a fundamentação teórica da pesquisa; a seção 3 aborda a metodologia empregada; nas seções 4 e 5 são descritos os resultados e o processo de validação.

2. Fundamentação Teórica

Para a fundamentação teórica foi utilizado um trabalho que comparou 4 diferentes ferramentas de simulação de SMAs.

2.1. Study and Comparison of Four Agent-Based Simulation Tools Repast, SeSAM, Netlogo and GAMA

O trabalho desenvolvido por [Mefteh 2014] teve como objetivo fazer a comparação das ferramentas de simulação baseadas em agentes Repast, SeSAM, NetLogo e Gama, de acordo com alguns critérios de avaliação não determinísticos, considerados importantes pelo autor, quando modelando Sistemas Multiagentes. A pontuação dos critérios pode ser 0, 20, 50 ou 100, sendo que quanto maior a pontuação, melhor a ferramenta é neste critério. A Tabela 1 mostra os critérios de avaliação pelos quais as ferramentas foram qualificadas.

Repast (http://repast.sourceforge.net/repast_3) foi desenvolvido na University of Chicago's Social Science Research Computing Lab, especificamente para criar simulações de ciências sociais. Contém muitos conceitos utilizados pelo Swarm, que foi a primeira ferramenta para modelagem e simulação que podia ser reutilizado para criar diferentes simulações. As simulações através do Repast possuem normalmente as classes Agent onde o agente é descrito, e a classe Model, que é responsável por executar e armazenar informações do modelo. A avaliação desta ferramenta se encontra na Tabela 2.

Sesam (<http://www.simsesam.de/>) é uma ferramenta genérica desenvolvida na Universidade de Wurzburg para a simulação de SMAs, com foco principal em construir modelos através de programação visual, sendo empregada em diversos projetos e em diferentes domínios. As principais entidades do Sesam são os *agents* que possuem variáveis e tem o comportamento descrito por diagramas de atividade, os *resources* que são parecidos com os *agents* porém não exercem comportamento e devem ser utilizados para entidades

passivas do modelo, e o *world* que representa o ambiente e toda a estrutura onde os agentes e *resources* ficarão localizados para a simulação. A avaliação desta ferramenta se encontra na Tabela 3.

NetLogo (<http://ccl.northwestern.edu/netlogo/>), criado em 1999 no Center for Connected Learning and Computer-Based Modeling na Tufts University, em Boston, como uma linguagem de programação para multiagentes e modelagem para simulação de fenômenos naturais e sociais. Nesta ferramenta, os agentes são chamados de Turtles e os recursos chamados de Patches. A avaliação desta ferramenta se encontra na Tabela 4.

Gama (<https://code.google.com/p/gama-platform/wiki/GAMA?tm=6>), desenvolvida por diversos pesquisadores da França e Vietnã no IRD/UPMC International Research Unit UMMISCO, visando entregar para especialistas de áreas, modeladores e cientistas da computação um ambiente de desenvolvimento de simulação e modelagem de sistemas multiagentes. O Gama é composto pelo agente global, *entities* que são as diferentes espécies que são usadas para definir a estrutura e comportamento dos agentes, o *Environment* que contém as definições dos ambientes, já que são suportados 3 diferentes topologias de ambientes e o *Experiment* que define experimentos para serem executados. A avaliação desta ferramenta se encontra na Tabela 5.

O **SIMULA** foi submetido aos mesmos critérios de avaliação que as ferramentas qualificadas na seção 2.1. A avaliação da ferramenta foi feita pelos autores deste trabalho. Ele foi desenvolvido na UFRGS como trabalho para obtenção do título de mestre por [Frozza 1997], e permite a criação de aplicações em sistemas multiagentes reativos. Por meio de uma interface gráfica, o usuário define sua aplicação e define um modelo, ou seja, a representação de um problema real, com de um grupo de agentes reativos que interagem entre si e com o ambiente no qual estão inseridos para atingir uma solução. A avaliação desta ferramenta se encontra na Tabela 6.

De acordo com os resultados coletados, o Sesam aparenta ser uma excelente ferramenta para simular Sistemas Multiagentes por ser uma ferramenta simples e livre que permite uma fácil implementação dos agentes. É através de um conjunto de atividades e transições entre essas atividades que o comportamento dos agentes, reativos e cognitivos, são implementados através de programação visual. Como ponto positivo tem-se o fato de ser possível alterar os dados dos agentes, do ambiente e dos recursos durante a simulação, possibilitando ao usuário interagir com a simulação em busca da convergência do comportamento do sistema para o que for adequado.

Através destas avaliações é possível concluir que o SIMULA é uma ferramenta forte e com grande capacidade de desenvolvimento e melhorias. Como pode ser visto na Tabela 7, onde foram agrupadas as avaliações das ferramentas, o SIMULA teve quatro critérios avaliados com pontuação máxima.

Tabela 1. Critérios para avaliação das ferramentas

Critério	Descrição do Critério	Valores Possíveis (V)	Taxa de Valor (V)
Disponibilidade	A ferramenta é livre e de código aberto?	V1 = livre e disponível. V2 = livre e não disponível.	V1 = 100. V2 = 0.
Atividade do Projeto	A ferramenta é utilizada em pesquisas?	V1 = Usado em pesquisas. V2 = Não usado em pesquisas.	V1= 100. V2 = 20.
Complexidade	Este critério avalia o nível de complexidade da ferramenta, incluindo a linguagem necessária para modelar e se há documentação suficiente.	V1 = Simples V2 = Médio V3 = Complexo	V1 = 100. V2 = 50. V3 = 0.
Generalidade	A ferramenta é genérica a ponto de ser possível simular problemas de diferentes domínios?	V1 = Especializada. V2 = Pouco especializada. V3 = Genérica.	V1 = 0. V2 = 20. V3 = 100.
Possibilidade de Uso de diagramas	Este critério permite definir se é possível modelar e atuar na simulação através de diagramas.	V1 = Com diagramas. V2 = Sem diagramas.	V1 = 100. V2 = 20.
Alterações em Simulação	Os parâmetros da simulação podem ser alterados durante a simulação?	V1 = Com alterações. V2 = Sem alterações.	V1 = 100. V2 = 0.
Métricas	Existem mecanismos de análise da simulação?	V1 = Com análise. V2 = Sem análise.	V1 = 100. V2 = 0.
Tipos de Agente	Quais os tipos de agentes? Reativos, Cognitivos ou ambos.	V1 = Reativos. V2 = Cognitivos. V3 = Ambos.	V1 = 0. V2 = 90. V3 = 100.

[Mefteh 2014]

Tabela 2. Descrição do Repast de acordo com os elementos de comparação.

Critério	Descrição do Critério	Pontuação
Disponibilidade	É livre e de código aberto	V1 = 100.
Atividade do Projeto	Está em desenvolvimento há 10 anos e utilizado por muitos pesquisadores.	V1= 100.
Complexidade	Suporta programação em Java. Os modelos podem ser desenvolvidos em diversas linguagens incluindo Java, C Sharp, C++, entre outras. Possui pouca documentação e poucos modelos de demonstração.	V3 = 0.
Generalidade	Foi feito para suportar principalmente o domínio de "ciências sociais" e inclui ferramentas específicas para este domínio.	V2 = 20.
Possibilidade de Uso de diagramas	Criação do comportamento de agentes em um diagrama representando suas propriedades e sequência de tarefas.	V1 = 100.
Alterações em Simulação	Possibilita acessar e modificar propriedades e comportamentos dos agentes, e propriedades do ambiente durante a execução.	V1 = 100.
Métricas	Não possui mecanismos de análise.	V2 = 0.
Tipos de Agente	Apenas agentes reativos.	V1 = 0.

[Mefteh 2014]

Tabela 3. Descrição do Sesam de acordo com os elementos de comparação.

Critério	Descrição do Critério	Pontuação
Disponibilidade	É livre e de código aberto.	V1 = 100.
Atividade do Projeto	Usado por pesquisadores e em aulas educacionais.	V1= 100.
Complexidade	Suporta programação visual, que é muito mais simples de aprender e usar do que linguagens de programação tradicionais. Provê extensa documentação, tutoriais e modelos de demonstração.	V1 = 100.
Generalidade	É uma plataforma de simulação genérica, não específica de apenas um domínio.	V3 = 100.
Possibilidade de Uso de diagramas	É possível descrever o comportamento do agente através de diagramas, fazendo a modelagem de agentes muito rápida e intuitiva.	V1 = 100.
Alterações em Simulação	Permite modificar o valor de um ou mais parâmetros em tempo de execução.	V1 = 100.
Métricas	Possui um analisador que monitora a simulação e apresenta os resultados em forma de gráficos ou armazenadas em forma de tabela em arquivo.	V1 = 100.
Tipos de Agente	Agentes reativos e cognitivos.	V3 = 100.

[Mefteh 2014]

Tabela 4. Descrição do Netlogo de acordo com os elementos de comparação.

Critério	Descrição do Critério	Pontuação
Disponibilidade	É livre e de código aberto.	V1 = 100.
Atividade do Projeto	Usado por milhares de estudantes, professores e pesquisadores ao redor do mundo.	V1= 100.
Complexidade	Suporta apenas programação em dialeto Logo. Diferente do Repast, Netlogo possui uma extensa documentação, tutoriais e uma biblioteca de modelos de demonstração que podem ser utilizados e modificados.	V2 = 50.
Generalidade	Foi feito como uma ferramenta educacional, com especialização em ciências sociais e naturais.	V2 = 20.
Possibilidade de Uso de diagramas	Não permite o uso de diagramas.	V2 = 20.
Alterações em Simulação	Não é possível alterar parâmetros em tempo de execução, a simulação deve ser parada para ser possível alterar.	V2 = 0.
Métricas	Não possui mecanismos de análise.	V2 = 0.
Tipos de Agente	Não podem ser chamados de agentes, apenas manipulação de objetos em um espaço bidimensional.	V1 = 0.

[Mefteh 2014]

Tabela 5. Descrição do Gama de acordo com os elementos de comparação.

Critério	Descrição do Critério	Pontuação
Disponibilidade	É livre e de código aberto.	V1 = 100.
Atividade do Projeto	É menos utilizado por pesquisadores do que as outras ferramentas comparadas. Possui extensa documentação e modelos de demonstração, além de vídeos de apresentação.	V1= 100.
Complexidade	Possui uma linguagem própria para definição de agentes, a GAML, que é simples de ser utilizada. Possui extensa documentação e modelos de demonstração.	V2 = 50.
Generalidade	Plataforma genérica.	V3 = 100.
Possibilidade de Uso de diagramas	Não é possível definir agentes através de diagramas, porém o ambiente e entidades da simulação podem ser definidas através de diagramas.	V2 = 20.
Alterações em Simulação	Permite a modificação de parâmetros de simulação em tempo de execução.	V1 = 100.
Métricas	Possui mecanismos de análise.	V1 = 100.
Tipos de Agente	Suporta agentes cognitivos.	V3 = 100.

[Mefteh 2014]

Tabela 6. Descrição do SIMULA de acordo com os elementos de comparação.

Critério	Descrição do Critério	Pontuação
Disponibilidade	É livre e de código aberto.	V1 = 100.
Atividade do Projeto	Voltou a ser desenvolvida (https://github.com/brunopick/SIMULA), e é utilizada por professores e alunos.	V1= 100.
Complexidade	Suporta programação visual, sendo muito mais fácil de aprender e utilizar do que linguagens de programação tradicionais. Possui documentação explicando seu funcionamento e modelos de demonstração.	V1 = 100.
Generalidade	É uma plataforma de simulação genérica, não específica a apenas um domínio.	V3 = 100.
Possibilidade de Uso de diagramas	Não permite o uso de diagramas.	V2 = 20.
Alterações em Simulação	Não é possível alterar parâmetros em tempo de execução.	V2 = 0.
Métricas	Não possui mecanismos de análise.	V2 = 0.
Tipos de Agente	Suporta agentes reativos.	V1 = 0.

(AUTORES, 2016)

Tabela 7. Comparação ferramentas

Critério	Repast	Sesam	Netlogo	Gama	SIMULA
Disponibilidade	V1 = 100	V1 = 100	V1 = 100	V1 = 100	V1 = 100
Atividade do projeto	V1 = 100	V1 = 100	V1 = 100	V1 = 100	V1 = 100
Complexidade	V3 = 0	V1 = 100	V2 = 50	V2 = 50	V1 = 100
Generalidade	V2 = 20	V3 = 100	V2 = 20	V3 = 100	V3 = 100
Possibilidade de uso de diagramas	V1 = 100	V1 = 100	V2 = 20	V2 = 20	V2 = 20
Alterações em simulação	V1 = 100	V1 = 100	V2 = 0	V1 = 100	V2 = 0
Métricas	V2 = 0	V1 = 100	V2 = 0	V1 = 100	V2 = 0
Tipos de agente	V1 = 0	V3 = 100	V1 = 0	V3 = 100	V1 = 0

(AUTORES, 2016)

3. Metodologia

Esta seção apresenta a metodologia utilizada para alcançar os objetivos deste trabalho.

3.1. Caracterização da pesquisa

O trabalho desenvolvido é caracterizado como uma pesquisa exploratória quanto ao objetivo, já que são pesquisados trabalhos relacionados para compreensão do domínio de estudo. Quanto ao objeto de estudo é uma pesquisa bibliográfica, com a busca de materiais em livros e artigos publicados na área por outros autores [GIL 2002].

O procedimento seguido é um estudo de caso, já que o trabalho foi desenvolvido utilizando o SIMULA, uma ferramenta já existente e validada. Quanto a sua natureza,

é uma pesquisa qualitativa, considerando-se que o principal objetivo é fazer o aprimoramento do ambiente [Yin 2015].

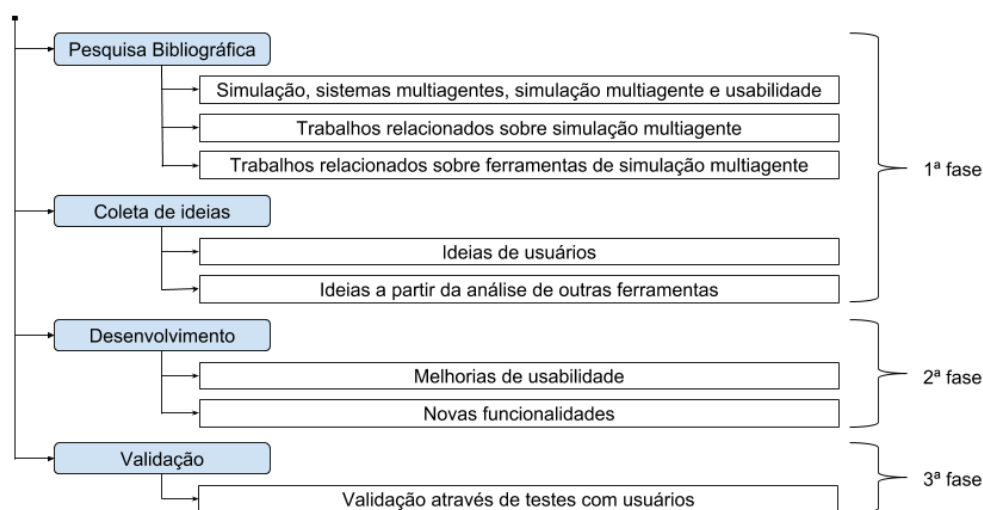
3.2. Etapas do desenvolvimento da pesquisa

A Figura 1 apresenta as etapas de desenvolvimento da pesquisa, que serão detalhadas nas próximas seções. A 1ª fase foi dividida em 2 partes. Na 1ª parte foi realizado um levantamento bibliográfico sobre simulação, sistemas multiagentes, simulação multiagente e usabilidade, depois foram levantados os trabalhos relacionados sobre simulação multiagente e ferramentas de simulação multiagentes. Na 2ª parte foi realizada a coleta de ideias a partir dos relatos de usuários do SIMULA e dos trabalhos relacionados sobre outras ferramentas. Com isso, foram propostas melhorias e novas funcionalidades para serem incorporadas ao SIMULA.

Depois, na 2ª fase foi iniciado o desenvolvimento, que foi dividido em 2 partes: melhorias de usabilidade e novas funcionalidades.

Na 3ª fase foram exibidos os resultados e foi realizada a validação das alterações através de testes com usuários.

Figura 1. Etapas de desenvolvimento da pesquisa.



(AUTORES, 2016)

4. Solução desenvolvida

Esta seção apresenta o desenvolvimento de novas funcionalidades e de melhorias às funcionalidades já existentes no SIMULA. A seção 4.1 apresenta o método para coleta das ideias de características que foram desenvolvidas e agregadas à ferramenta e a seção 4.2 demonstra o que foi desenvolvido.

4.1. Coleta de ideias

O ambiente SIMULA possui diversos pontos fortes, tais quais as ferramentas comparadas. A ferramenta é utilizada por diversas pessoas com níveis de conhecimento distintos, e justamente por esse motivo o método de coleta de novas funcionalidades e melhorias ter

sido definido através das ideias que os próprios interessados solicitam é muito benéfica, já que são pessoas que por usarem a ferramenta observam certos pontos que podiam ser melhorados ou adicionados.

Como o SIMULA é utilizado no meio acadêmico, muitas ideias surgiram dos próprios usuários (estudantes e professores) durante o uso da ferramenta. Essas ideias eram registradas no repositório do código fonte do sistema. Não foi feito um teste diretamente com usuários antes do desenvolvimento do trabalho, mas as propostas de ideias vieram das experiências do uso do SIMULA por parte dos usuários. Além das ideias sugeridas por usuários do SIMULA, outras ferramentas possuem algumas funcionalidades que não são encontradas no SIMULA, assim é possível analisar as mesmas e aplicar ao ambiente aquelas que são pertinentes. O SIMULA tem sido utilizado em sala de aula faz anos e as sugestões sempre armazenadas em arquivos, como uma lista sendo atualizada a cada semestre. As sugestões consideradas foram coletadas nos últimos cinco anos.

4.2. Desenvolvimento

A partir de ideias de diversos usuários do SIMULA e melhorias advindas a partir da análise de outras ferramentas, os itens coletados que foram implementados no SIMULA serão apresentados nesta seção.

As novas funcionalidades desenvolvidas nesse trabalho foram: Deixando pistas; Movendo agentes; Excluindo agentes; Log da simulação; Evitando o embolo; Permitir controlar o tempo de cada ciclo.

As melhorias de usabilidade implementadas na ferramenta foram: Alterar *LookAndFeel* para melhorar *interface*; Palheta de cores; Permitir desenhar o agente arrastando o mouse; Selecionar imagem no Agente; Não permitir caracteres especiais e espaço em branco no nome de agentes e variáveis; Duplicar definição de regra de agente; Ter uma versão de fácil instalação (release); Modelo só executa no mesmo diretório do SIMULA.

4.3. Considerações

Com o desenvolvimento das melhorias "Deixando pistas", "Excluindo agentes", "Movendo agentes", "Evitando o embolo" e "Controle de tempo por ciclo" é possível fazer alterações diretamente na simulação sem ser necessário alterar o modelo. Através destas funcionalidades o usuário consegue interagir com a simulação em tempo real, e com isso abrir novas possibilidades na simulação, como por exemplo verificar o comportamento de um agente ao movê-lo para outra posição do ambiente. Ainda na etapa de simulação, por meio do "Log da simulação" que foi criado é possível analisar cada ciclo da simulação minuciosamente e quantas vezes for necessário. A característica da interatividade é o ponto principal para gerar diferentes situações e configurações dos agentes e sua atuação na simulação multiagente, a fim de aproximar a simulação da realidade modelada.

A ferramenta também recebeu diversas melhorias de usabilidade, com o desenvolvimento de "Desenhar agente arrastando o mouse", "Duplicar regra de agente", "Selecionar imagem no agente", "Não permitir caracteres especiais e espaço em branco no nome de agentes e variáveis", "Palheta de cores", "Alterar *LookAndFeel* para melhorar *interface*". Essas melhorias foram desenvolvidas para facilitar o uso da ferramenta, como por exemplo a duplicação de regra de agente, pois com ela o usuário consegue mais agilidade para criar regras que são muito parecidas, e o controle de caracteres especiais no nome

do agente, já que sem isso muitas vezes ocorriam problemas que não eram facilmente detectados na modelagem.

Ao relacionar as 5 características de usabilidade de Nielsen:2007 com o que foi desenvolvido, é possível afirmar:

- **Aprendizagem:** as alterações de salvar e da definição de agente ajudam na aprendizagem pois estão mais simples.
- **Eficiência:** a característica de interatividade na simulação ajuda na eficiência pois não é preciso sair da simulação para fazer alguns tipos de alteração na mesma, já o *log* ajuda a analisar a simulação posteriormente e modificá-la se for preciso.
- **Facilidade de memorização:** a simplicidade do SIMULA e baixo conhecimento necessário fazem a ferramenta ser facilmente memorizada.
- **Segurança:** não ser permitido colocar caracteres inválidos em nomes de agentes e variáveis e corrigir a ação de salvar são alguns pontos que melhoram a segurança, pois evita que aconteçam erros.
- **Satisfação:** a possibilidade de efetuar alterações em tempo de simulação torna a tarefa de modelagem e análise da simulação mais agradável.

5. Processo de avaliação

Os testes e tarefas que foram realizados no sistema é chamada de Lista de Tarefas [Ferreira and da Silva 2002]. Apesar da prévia leitura desta lista ser opcional aos participantes, podendo estes lerem apenas no momento da execução dos testes, recomenda-se que leiam antes das tarefas serem executadas, pois com isso o participante estará familiarizado com a tarefa e se sentirá mais confortável. As cinco tarefas executadas pelos usuários foram, na ordem: Definição de agentes; Definição de variáveis; Definição de critério de parada; Definição de regras de comportamento; Executar simulação. Uma simulação por completa foi apresentada aos usuários para que executassem as tarefas solicitadas.

5.1. Questionário de Avaliação do Sistema

Após a execução da lista de tarefas os participantes preencheram um questionário, com a finalidade de extrair informações para elucidar e observar o entendimento do sistema [Ferreira and da Silva 2002]. O participante pode informar, através deste questionário, os pontos positivos e negativos do sistema, embasado em sua experiência real com o sistema. Para isso é necessário perguntas formuladas que estão diretamente relacionadas com aspectos como sentimentos, opiniões e sugestões de melhoria [Barbosa and Silva 2010]. Neste sentido as perguntas devem ser claras, de forma que as respostas sejam simples e concisas para facilitar a análise dos dados coletados.

5.2. Método

Os testes foram realizados pelos participantes entre os dias 10 e 12 de novembro de 2016, no lugar de sua escolha, já que os materiais foram enviados por e-mail. No total foram seis participantes, que possuíam conhecimento em sistemas multiagentes e haviam utilizado o SIMULA sem as alterações implementadas por este trabalho. Os participantes inicialmente executaram a Lista de Tarefas na ferramenta e após isso responderam ao Questionário.

5.3. Resultados

Esta seção apresenta os resultados dos testes realizados com seis usuários.

5.3.1. Respostas ao Questionário de Avaliação do Sistema

A Tabela 8 apresenta uma avaliação das respostas dos participantes do teste ao questionário de avaliação do sistema onde as respostas são oferecidas ao participante em uma escala 1 a 5. Esta avaliação realizada pelos usuários foi importante para observar se as alterações deste trabalho cumpriram o seu objetivo.

Tabela 8. Respostas objetivas dos participantes.

Questão / Usuário	1	2	3	4	5	6	Média
Facilidade de utilização	3	4	4	2	3	5	3,50
Organização das informações	1	5	5	4	4	5	4,00
Layout das telas	2	5	5	4	4	5	4,16
Nomenclatura utilizada nas telas	4	5	5	3	4	5	4,33
Mensagens do sistema	3	4	3	5	5	5	3,33
Assimilação das informações	4	4	5	4	4	5	4,33
No geral, a realização do teste foi	4	5	5	1	5	5	4,16

(AUTORES, 2016)

Uma síntese das respostas dos usuários para as questões subjetivas do Questionário de Avaliação do Sistema encontram-se a seguir:

1. Aponte situações que você achou fácil utilizar no sistema: Criação e definição dos agentes, execução e controle da simulação.
2. Aponte situações que você sentiu dificuldades: Definição de regras de comportamento.
3. Diante do teste realizado, você acha que as melhorias implementadas no SIMULA atingiram o objetivo para o qual foram desenvolvidas? Sim, a criação do agente está mais fácil e agora é possível controlar melhor a simulação.
4. O espaço a seguir é reservado para que você exponha sua opinião e sugira melhorias no sistema: Apesar das melhorias gerais na ferramenta a definição das regras de comportamento ainda é difícil e pouco dinâmica, sendo uma tarefa monótona.

6. Conclusão

Das questões objetivas as duas com a menor média são "Facilidade de utilização" e "Mensagens do sistema", com notas 3,5 e 3,33, respectivamente. Ao analisar foi verificado que essas médias baixas se dão ao fato de que ambas estão envolvidas com a criação das Regras de Comportamento.

De acordo com as respostas dos usuários todas as melhorias desenvolvidas atingiram seu objetivo e ajudaram de alguma forma a experiência dos mesmos no SIMULA. Ao analisar as respostas para a Questão "Aponte situações que você sentiu dificuldades." foi unânime o sentimento de dificuldade na tela das Regras de Comportamento, que não foi alterada no presente trabalho. Como a dificuldade dos usuários ficou na criação das regras de comportamento, estes opinaram que esta parte deveria ser melhorada, e foi quase

unânime a opinião de que deveria ser possível alterar qualquer posição da regra sem ser necessário apagar parte dela, pois para alterar algo no início da regra é necessário apagar boa parte da mesma.

O item "Alterações em Simulação" teria a pontuação aumentada de "V2 = 0" para "V1 = 100", e o item "Métricas" seria alterado de "V2 = 0" para "V1 = 100". Isso se deve ao fato de que foram incorporadas diversas melhorias na simulação e na ferramenta em geral. Com isso a comparação das ferramentas fica como pode ser visto na Tabela 9, onde as alterações estão destacadas.

Tabela 9. Comparação ferramentas após validação.

Critério	Repast	Sesam	Netlogo	Gama	SIMULA
Disponibilidade	V1 = 100	V1 = 100	V1 = 100	V1 = 100	V1 = 100
Atividade do projeto	V1 = 100	V1 = 100	V1 = 100	V1 = 100	V1 = 100
Complexidade	V3 = 0	V1 = 100	V2 = 50	V2 = 50	V1 = 100
Generalidade	V2 = 20	V3 = 100	V2 = 20	V3 = 100	V3 = 100
Possibilidade de uso de diagramas	V1 = 100	V1 = 100	V2 = 20	V2 = 20	V2 = 20
Alterações em simulação	V1 = 100	V1 = 100	V2 = 0	V1 = 100	V1 = 100
Métricas	V2 = 0	V1 = 100	V2 = 0	V1 = 100	V1 = 100
Tipos de agente	V1 = 0	V3 = 100	V1 = 0	V3 = 100	V1 = 0

(AUTORES, 2016)

Referências

- Barbosa, S. D. J. and Silva, B. S. d. (2010). *Interação humano-computador*. Elsevier Brasil.
- Ferreira, K. G. and da Silva, C. I. P. P. (2002). *Teste de Usabilidade*. UFMG, Dissertação (Bacharelado). Belo Horizonte, MG.
- Frozza, R. (1997). Simula - ambiente para desenvolvimento de sistemas multiagentes reativos. Master's thesis, CPGCC/UFRGS, Porto Alegre, RS. 117f.
- GIL, A. C. (2002). *Modelagem empresarial: ferramentas para tomada de decisão*. Atlas, São Paulo, SP.
- Krug, S. (2006). *Não me faça pensar!: uma abordagem de bom senso à usabilidade na web*. Alta Books, Rio de Janeiro, RJ.
- Mefteh, W. (2014). Study and comparison of four agent-based simulation tools: Repast, sesame, netlogo and gama. *International Journal of Computers & Technology*, 13(4).
- Yin, R. K. (2015). *Estudo de Caso: Planejamento e Métodos*. Bookman, Porto Alegre, RS.

Formação do Consenso por meio de Diálogos Argumentativos em Sistemas Multiagentes

Ayslan T. Possebom¹, Mariela Morveli-Espinoza², Everson Joay², Cesar A. Tacla²

¹Instituto Federal do Paraná câmpus Paranavaí (IFPR)
R. José Felipe Tequinha 1400, Jardim das Nações - 87703-536 – Paranavaí, PR – Brasil

²Pós-Graduação em Engenharia Elétrica e Informática Industrial
Universidade Tecnológica Federal do Paraná (UTFPR) – Curitiba, PR - Brasil
{possebom,morveli.espinoza,eversonjoay}@gmail.com, tacla@utfpr.edu.br

Abstract. *An intelligent agent may have multiple beliefs consisting of facts and inference rules in his knowledge base. These beliefs can be used to build arguments in favor or against any decision alternative or other arguments, which leads to some conflict of opinions among the agents. In this paper, we focus on these conflicts among beliefs. Our approach is based on a dialogue where agents can exchange arguments about some issue and the beliefs that are most agreed by the majority are consented by the group.*

Resumo. *Um agente inteligente pode ter múltiplas crenças em sua base de conhecimento, sendo formadas por fatos ou regras de inferência. Estas crenças podem ser usadas para construir argumentos apoiando ou rejeitando uma alternativa de decisão ou outros argumentos, podendo gerar algum tipo de conflito de opiniões entre os agentes. Neste artigo, focamos nestes conflitos entre crenças. Nossa abordagem é baseada em um diálogo onde os agentes podem trocar argumentos sobre um determinado assunto e as crenças que são mais apoiadas pela maioria dos agentes são consentidas pelo grupo.*

1. Introdução

O termo consenso se refere ao princípio de tomada de decisão onde um grupo de pessoas ou agentes devem entrar em algum tipo de acordo coletivo. Este tipo de tomada de decisão requer uma unanimidade, ou seja, todos os membros do grupo (ou no pior caso, a maioria dos membros) devem concordar com a mesma decisão e motivos que justifiquem a sua escolha. No entanto, o processo de decisão por consenso em um grupo de pessoas não é uma tarefa trivial, visto que as pessoas devem possuir algum tipo de objetivo em comum, não deve existir competição entre pontos de vistas individuais, deve haver uma participação ativa dos membros e também deve existir tempo suficiente para os debates [Hertzberg, 2003].

Os métodos de tomada de decisão em grupo mais realísticos entre pessoas, tais como eleições [Romme, 2004], bem como suas variações como Borda, Condorcet e Agregação de Julgamentos [Grossi e Pigozzi, 2014], não permitem que existam diálogos entre os membros do grupo. Cada participante apresenta seu voto a favor ou contra uma determinada alternativa de decisão e o resultado é computado com base na quantia de votos recebidos, podendo indicar qual a alternativa preferida pelo grupo ou a ordem de preferência entre as alternativas. Estes métodos não permitem aos membros explicarem

as razões do seu voto, nem a possibilidade dos demais membros mudarem de opinião no decorrer da votação. Neste sentido, a aplicação do diálogo onde todos os participantes possam emitir suas opiniões se torna um passo prévio importante antes da votação.

Em um sistema multiagente, este diálogo com mensagens contendo as opiniões ou motivos pelos quais os agentes apoiam ou rejeitam as alternativas de decisões, bem como outras opiniões, pode ser feito por meio de um sistema de argumentação. Diversos trabalhos aplicados à tomada de decisões utilizando sistemas de argumentação foram propostos, tais como geração de argumentos a favor ou contra uma alternativa de decisão [Amgoud et al., 2005], uso de argumentos epistêmicos e práticos [Amgoud e Prade, 2009], relação entre atributos de decisão com alternativas de decisão e objetivos dos agentes [Zhong et al., 2014] ou votação a favor ou contra cada argumento fornecido [Leite e Martins, 2010], entre outros. Estes trabalhos aplicam um princípio de decisão em um único agente para a escolha do melhor argumento a ser emitido, ou então, tentam estabelecer uma ordem de preferência dos argumentos apresentados.

Com relação ao consenso, precisamos analisar a estrutura interna de cada argumento, considerando o quanto cada informação do argumento é conhecida ou não pelos demais agentes do sistema, ou seja, determinar o que é conhecimento comum. Argumentos com informações que são amplamente conhecidas tendem a ser mais fortes que os demais, sofrendo menor número de ataques. A formação do consenso em sistemas argumentativos pode ser útil em uma série de domínios de aplicação, como chatbot, redes de sensores, classificação de importância de páginas web, formação de opiniões, etc.

Este trabalho tem como objetivo propor um mecanismo de diálogos em um sistema multiagente onde os argumentos são avaliados pelo grupo de agentes na tentativa de identificar o quanto cada informação em um argumento é considerada aceita pelo grupo (nível de consenso). Cada argumento recebe uma força que é baseada no nível de consenso de suas fórmulas. Esta força representa o quanto um argumento é aceito pelo grupo de agentes quanto comparado com os demais argumentos apresentados. Como resultado do diálogo, um grafo ponderado de argumentos é criado e pode ser usado para a seleção da alternativa que mais é sustentada pelo grupo de agentes.

A próxima seção apresenta o framework proposto para a formação do consenso e construção do conhecimento comum em um diálogo argumentativo. A seção 3 é dedicada a aplicação do modelo de diálogo de um grupo de agentes sobre um assunto ou alternativa de decisão. A Seção 4 apresenta as características do modelo com relação ao consenso. Finalmente, a seção 5 traz as conclusões e trabalhos futuros.

2. Formação do consenso e Conhecimento Comum

O modelo proposto para a construção do conhecimento comum e a formação do consenso é baseado na comunicação de um grupo de agentes, onde cada mensagem enviada representa um argumento e cada argumento é avaliado pelo grupo por meio de votação de aceitação ou rejeição em suas fórmulas. Cada agente do sistema possui um valor que representa sua confiabilidade ou *expertise* para o grupo de agentes. Este valor de *expertise* representa o quanto cada informação em um argumento deve ser considerado como aceito pelo grupo.

Definição 1 (FCC): Um Framework para a construção do Conhecimento Comum (FCC) é formado por uma tupla $\langle \mathcal{AG}, \mathcal{TS}, med, \mathcal{D}, \tau \rangle$ onde: $\mathcal{AG} = \{ag_1, \dots, ag_n\}$ com $n > 1$ é o conjunto de agentes argumentadores, $\mathcal{TS} = \{ts_1, \dots, ts_n\}$ com $\{ts_i \in \mathbb{R} | 0 \leq ts_i \leq 1\}$ e

$ts_1 + \dots + ts_n = 1$ é o conjunto de valores de expertise dos agentes argumentadores onde ag_i possui expertise ts_i , med é o agente mediador responsável por controlar a sequência de mensagens e votação, \mathcal{D} é o conjunto de alternativas de decisões a serem discutidas pelo grupo de agentes com $\mathcal{D} = \{d_1, \dots, d_m\}$ e $m \geq 1$, e t representa um tempo de espera em milissegundos que o agente mediador utiliza para sincronizar as mensagens e votos recebidos durante o diálogo.

Os valores de cada expertise são definidos no framework, no entanto, podem resultar de sistemas de reputação e confiança. O framework utiliza uma normalização dos valores de expertise, permitindo que o consenso máximo no grupo de agentes seja 1. Como o processo de tomada de decisão envolve um grupo específico de agentes, a aplicação do framework se limita a sistemas multiagentes fechados.

Definição 2 (agente argumentativo): Um agente argumentativo $ag_i \in \mathcal{AG}$ é formado por um par $\langle \Sigma_i, \mathcal{CS}_i \rangle$ onde $\Sigma_i = K_i \cup KO_i$ forma sua base de conhecimentos com K_i representando suas crenças sobre o ambiente e KO_i as crenças obtidas na comunicação do agente ag_i com $\mathcal{AG} \setminus ag_i$, e $\mathcal{CS}_i = \mathcal{S}_i \cup \mathcal{A}_i$ forma a base de armazenamento de compromissos do agente com \mathcal{S}_i representando um conjunto de argumentos a serem enviados e \mathcal{A}_i um conjunto unitário que armazena o argumento que será utilizado para identificar possíveis contra-argumentos.

Um argumento emitido por um agente representa sua opinião ou ponto de vista sobre a alternativa de decisão em discussão, ou então, um contra-argumento. Neste trabalho, uma base de conhecimentos Σ é formada por fórmulas representando crenças sobre fatos e regras em uma linguagem proposicional e os argumentos são construídos com base nestas fórmulas. Os símbolos \vdash , \equiv e \perp representam inferência clássica, equivalência e contradição, respectivamente. Um argumento consiste em um par $\langle \Phi, \alpha \rangle$ onde Φ representa o conjunto de fórmulas (ϕ_1, ϕ_2, \dots) da premissa de um argumento e α sua conclusão, onde (1) α é uma fórmula, (2) $\Phi \subseteq \Sigma$, (3) $\Phi \not\vdash \perp$, (4) $\Phi \vdash \alpha$, (5) $\nexists \Phi' \subseteq \Phi$ tal que $\Phi' \vdash \alpha$ [Besnard & Hunter, 2014]. As relações de ataque entre argumentos utilizadas para a construção de contra-argumentos são *undercut* e *rebuttal* [Parsons & McBurney, 2003]. Sejam dois argumentos distintos $arg_1 = \langle \Phi_1, \alpha_1 \rangle$ e $arg_2 = \langle \Phi_2, \alpha_2 \rangle$, dizemos que um arg_1 faz um ataque por *undercut* em arg_2 se $\exists \phi \in \Phi_2$ tal que $\alpha_1 \equiv \neg \phi$, e um arg_1 faz um ataque por *rebuttal* em arg_2 se $\alpha_1 \equiv \neg \alpha_2$. A notação $arg_i \mathcal{R} arg_j$ representa que arg_i faz um ataque (*undercut* ou *rebuttal*) a arg_j .

Definição 3 (agente mediador): Um agente mediador med é formado por uma tupla $\langle wb, Agenda, dt, Bel \rangle$ onde wb (*whiteboard*) representa uma lista de agentes argumentadores inscritos para enviar argumentos, $Agenda$ contém uma lista de argumentos enviados por um agente em um dado instante, dt (*dialogue table*) contém todos os argumentos apresentados pelo grupo de agentes durante o diálogo, e Bel é a base de conhecimento do agente mediador formada pelas crenças (fatos e regras) contidos nos argumentos em dt .

O wb consiste em uma lista que funciona como um mecanismo para emular o que ocorre em reuniões presenciais. Quando um agente ag_i deseja enviar algum argumento, ele solicita sua inscrição nesta lista e aguarda por sua vez. Esta lista garante que apenas os agentes inscritos recebam o direito de enviar argumentos de forma coordenada. Um agente pode ocupar apenas uma posição no wb em um dado instante. Apenas o agente

ocupante da primeira posição na lista recebe o direito de emitir argumentos em cada turno de falas.

Quando um agente ag_i recebe o direito de fala, ele envia para med todos os argumentos em \mathcal{S}_i e então $\mathcal{S}_i = \emptyset$. O agente med armazena os argumentos recebidos na *Agenda* para serem verificados. A verificação dos argumentos se torna necessária para que todos os argumentos estejam relacionados à discussão (ex.: tenham alguma relação de ataque com os argumentos já apresentados), para que eles ainda não tenham sido apresentados até o momento (ex.: evitando repetições), e para que contenham fórmulas que possam ser aceitas pelo grupo (ex.: uma nova fórmula ainda não apresentada até o momento ou uma fórmula já apresentada anteriormente e aceita pelo grupo).

A tabela de diálogos dt é populada com os argumentos válidos da *Agenda*. Cada linha desta tabela contém um número sequencial de identificação do argumento, o agente que apresentou o argumento, o argumento, e uma listagem das fórmulas do argumento com os votos de apoio ou rejeição recebidos.

Sempre que med insere um novo argumento arg_j em dt , ou seja, $dt = dt \cup arg_j$, temos que $\forall ag_i \in \mathcal{AG}, \mathcal{A}_i = arg_j$. Quando o grupo de agentes é solicitado a atacar arg_j , os ataques encontrados são armazenados em \mathcal{S}_i e então \mathcal{A}_i é anulada para cada agente. Portanto, se $\exists arg_k | arg_k \mathcal{R} arg_j$ então $\forall arg_k, \mathcal{S}_i = \mathcal{S}_i \cup arg_k$ e $\mathcal{A}_i = \emptyset$.

Todo argumento enviado em um diálogo passa por um processo de votação de apoio ou rejeição em cada fórmula deste argumento. Este processo visa a identificação do nível de consenso do grupo sobre cada fórmula (informação) presente no argumento.

Seja $ARG(\Sigma)$ o conjunto de todos os argumentos que podem ser construídos a partir de Σ , arg é um argumento enviado por um agente, η é uma fórmula em arg , μ é uma fórmula em Σ , $premise : arg \rightarrow \Phi$ uma função que retorna o conjunto de fórmulas da premissa de um argumento, $claim : arg \rightarrow \alpha$ uma função que retorna a conclusão de arg , $atoms : \eta \rightarrow \Pi$ uma função que retorna as átomos de η . Uma fórmula deste argumento $\eta \in premise(arg) \cup claim(arg)$ pode receber votos de apoio ou rejeição de outros agentes.

Definição 4 (voto de apoio): Uma fórmula η de um argumento enviado pelo agente ag_i será apoiada por ag_j se e somente se:

1. $\exists arg_2 \in ARG(\Sigma_j) | claim(arg_2) = \eta$, ou;
2. $\exists \mu \in \Sigma_j | atoms(\mu) = atoms(\eta)$ e $\mu \leftrightarrow \eta$ é uma tautologia.

Definição 5 (voto de rejeição): Uma fórmula η de um argumento enviado pelo agente ag_i será rejeitada por ag_j se e somente se:

1. $\exists \mu \in \Sigma_j | atoms(\mu) = atoms(\eta)$ e $\mu \leftrightarrow \neg \eta$ é uma tautologia.

Após a fase de votação em cada fórmula do argumento, os agentes podem atualizar suas bases de conhecimentos Σ_i para todo $ag_i \in \mathcal{AG}$ com as fórmulas que foram consideradas aceitas pelo grupo. Para determinar quando uma fórmula deverá ser considerada aceita, utilizamos uma função de atualização de crenças $fac : \eta \rightarrow \mathbb{R}$ para calcular o nível de consenso. A Equação 1 mostra como o valor desta função é obtido. Representamos por ag_i o agente emissor do argumento, $Support[\eta]$ (e $Reject[\eta]$) o

conjunto de agentes que apoiam (e rejeitam, respectivamente) a fórmula η durante a fase de votação.

$$fac(\eta) = ts_i + \sum_{ag_j \in Support[\eta]} ts_j - \sum_{ag_j \in Reject[\eta]} ts_j \quad (1)$$

A função fac é utilizada por med para determinar se o grupo deverá considerar η ou $\neg\eta$ como fórmula consensualmente aceita utilizando o conceito de voto majoritário baseando-se no valor de $expertise$ dos agentes. Utilizamos um valor limiar σ para determinar um valor limite mínimo para que a fórmula seja considerada aceita pelo grupo. Este valor é definido em tempo de projeto. Quanto maior o valor de σ , mais cético será o grupo de agentes. O resultado pode ser obtido com:

$$fac(\eta) \begin{cases} > \sigma & \Sigma = \Sigma \cup \eta \\ < -\sigma & \Sigma = \Sigma \cup \neg\eta \end{cases}$$

Sempre que o resultado de fac for maior que o limiar, a fórmula η do argumento deverá ser considerada aceita pelo grupo. Valores entre $[-\sigma, \sigma]$ indicam que não existe consenso do grupo sobre a fórmula η . Quando o resultado de fac for inferior a $-\sigma$, o grupo de agentes deverá considerar a negação da fórmula como aceita.

Fórmulas que são consensualmente aceitas devem ser incorporadas na base de conhecimentos do agente. Um rótulo deve ser incorporado a esta fórmula indicando para qual alternativa de decisão esta informação foi considerada aceita pelo grupo, juntamente com o valor do nível de consenso. Representamos as fórmulas consensualmente aceitas pelo grupo por $\eta[d_i(fac)]$, onde d_i é uma alternativa de decisão. Fórmulas na base de conhecimentos que possuem anotações indicam que esta informação já foi apresentada anteriormente em algum diálogo e foi consensualmente aceita pelo grupo para uma determinada alternativa de decisão. Todas as fórmulas de um argumento são inseridas em Bel com suas respectivas anotações. O agente med utiliza esta base para fazer a verificação da validade dos argumentos.

Quando a fórmula aceita pelo grupo está presente na base K de um agente, apenas o rótulo é atualizado nesta fórmula. Quando a fórmula aceita pelo grupo não faz parte da base de conhecimentos do agente, a base KO deverá então ser atualizada para incorporar a nova informação. Desta forma, a base KO será incrementada com informações desconhecidas pelo agente. Neste sentido, as fórmulas em K representam informações que o agente possui sobre o ambiente e serão sempre verdadeiras para este agente, KO representa informações aprendidas durante o diálogo e serão sempre aceitas para o agente, com exceção de situações onde a base de dados se torna inconsistente em K e KO . Neste caso, o agente continua acreditando em suas crenças em K para diálogos sobre outras alternativas de decisão e considera as crenças em KO apenas em diálogos sobre a alternativa de decisão contida no rótulo da crença, visto que para aquela alternativa de decisão, a crença em KO foi preferida pelo grupo.

Exemplo 1: Seja $\mathcal{AG} = \{ag_1, ag_2, ag_3, ag_4, ag_5, ag_6\}$, $\mathcal{TS} = \{0.22, 0.39, 0.09, 0.09, 0.04, 0.17\}$ e $\mathcal{D} = \{d_1\}$. Em algum momento durante o diálogo sobre a alternativa de decisão d_1 com $\sigma = 0.1$, o agente ag_5 envia o argumento $\langle\{a, a \rightarrow b\}, b\rangle$ atacando outro argumento que afirma b .

- Considerando $Support[a] = \{ag_1, ag_3, ag_6\}$ e $Reject[a] = \{ag_2\}$, temos que $fac(a) = 0,13$ e $fac(a) > \sigma$ indicando que a fórmula $\eta = a$ foi

consensualmente aceita pelo grupo. Agentes que não conhecem esta fórmula (ag_4) e os que a rejeitam (ag_2) devem aceitar a fórmula a . Desta forma, a base KO de ag_2 e ag_4 incorporará a fórmula $a[d_1(0,13)]$ e os agentes ag_1 , ag_3 , ag_5 e ag_6 que tiverem a fórmula a na base K devem atualizá-la, incorporando o rótulo;

- Considerando $Support[a \rightarrow b]=\{\}$ e $Reject[a \rightarrow b] = \{ag_3, ag_4\}$, temos que $fac(a \rightarrow b) = -0,14$ e $fac(a \rightarrow b) < -\sigma$ indicando que a fórmula $\eta = \neg(a \rightarrow b)$ foi consensualmente aceita pelo grupo. Agentes que não conhecem esta fórmula (ag_1, ag_2, ag_6) e os que a apoiam (ag_5) devem aceitar a fórmula $\neg(a \rightarrow b)$. Desta forma, a base KO ou K dos agentes devem conter $\neg(a \rightarrow b)[d_1(0,14)]$;
- Considerando $Support[b]=\{\}$ e $Reject[b] = \{ag_2, ag_3, ag_4\}$, temos que $fac(b) = -0,53$ e $fac(b) < -\sigma$ indicando que a fórmula $\eta = \neg b$ foi consensualmente aceita pelo grupo. Agentes que não conhecem a fórmula e os que a apoiam devem aceitar $\neg b$. Desta forma, a base KO ou K dos agentes devem conter $\neg b[d_1(0,53)]$. O tratamento desta inconsistência é apresentado na seção 4.

O agente *med* é o responsável por conduzir o processo de diálogo entre os agentes argumentadores. Este agente cria a tabela de diálogos, a agenda e a lista de inscrições. Inicialmente o mediador cria um argumento para cada alternativa de decisão. Estes argumentos são inseridos na tabela de diálogos e informados ao grupo de agentes argumentadores. Os agentes buscam por contra-argumentos e votam nas fórmulas dos argumentos em um processo iterativo. Uma proposta de modelo de diálogos para a construção do conhecimento comum entre um grupo de agentes argumentativos é apresentado no Algoritmo 1. As funções definidas no Algoritmo são:

createWhiteboard(), **createAgenda()** e **createDialogueTable()**: o agente *med* inicializa as estruturas que serão utilizadas durante o processo de diálogo para registrar agentes que desejam expor argumentos, armazenar os argumentos enviados pelos agentes e armazenar os argumentos válidos utilizados no diálogo.

createInitialArgument(d): agente *med* cria argumento para cada alternativa de decisão. Ex.: $\langle \{d_i\}, d_i \rangle$;

dtAddArgument(DT, ag , arg , $ataque$): agente *med* insere uma nova linha na tabela de diálogos contendo o agente emissor do argumento, o argumento enviado e as relações de ataque (ligação com outros argumentos);

broadcastArgument(arg , y): agente *med* informa aos agentes argumentadores que existe um argumento em análise. Este argumento será incluído na base de armazenamento de compromissos para análise \mathcal{A} dos agentes argumentadores. O parâmetro y representa a linha na tabela que contém o argumento;

broadcastAttack(hora atual, t): agente *med* informa aos argumentadores que eles devem buscar por contra-argumentos em um tempo t definido no FCC . Os argumentos encontrados são registrados no armazenamento de compromissos para envio S dos agentes e \mathcal{A} é esvaziado;

broadcastAskSpeak(): agente *med* solicita ao grupo quais agentes desejam emitir argumentos. Agentes que possuem $S \neq \emptyset$ respondem solicitando a inscrição na wb ;

addWhiteboard(wb , $agentes$): agente *med* atualiza a listagem dos agentes que desejam falar adicionando os agentes que solicitaram a permissão de fala (**broadcastAskSpeak**) e ainda não estão presentes na lista;

Algoritmo 1: Processo de Diálogo para Construção do Conhecimento Comum

```

1  procedure DIALOGUE
2    wb ← createWhiteboard()
3    dtm ← createDialogueTable()
4    agenda ← createAgenda()
5    for each decision in  $\mathcal{D}$ 
6      initArg ← createInitialArgument(decision)
7      y ← dtAddArgument(dtm, med, initArg, null)
8      broadcastArgument(initArg, y)
9      Agree ← broadcastAskVotingAgreement(decision)
10     b ← fac(null, Agree, null)
11     dtUpdate(dtm, y, decision, Agree, null, b)
12     dtUpdate(dtm, y, is(initArg))
13     broadcastAttack(currentTime, t)
14     wait(t)
15     addWhiteboard(wb, broadcastAskSpeak())
16     while (!emptyWhiteboard(wb))
17       currentAgent ← getWhiteboard(wb, 0)
18       listArgs ← askArguments(currentAgent)
19       fillAgenda(agenda, listArgs)
20       validateArguments(agenda)
21       for each a in agenda do
22         y ← dtAddArgument(dtm, currentAgent, a, getAttack(a))
23         broadcastArgument(a, y)
24         formulasArgument ← split(a)
25         for each f in formulasArgument do
26           Agree ← broadcastAskVotingAgreement(f)
27           Reject ← broadcastAskVotingRejection(f)
28           b ← fac(currentAgent, Agree, Reject)
29           dtUpdate(dtm, y, f, Agree, Reject, b)
30           if b >  $\sigma$  then broadcastLearn(f, decision, b)
31           else if b <  $-\sigma$  then broadcastLearn(!f, decision, b+1)
32           else broadcastLearn(f, decision, 0)
33         end for
34         broadcastAttack(currentTime, t)
35         wait(t)
36       end for
37       updateWhiteboard(wb)
38       addWhiteboard(wb, broadcastAskSpeak())
39     end while
40   end for
41 end procedure

```

updateWhiteboard(wb): agente *med* atualiza a lista de inscrições de fala, removendo o agente na primeira posição (que já apresentou seus argumentos), tornando o próximo agente inscrito habilitado a falar;

broadcastAskVotingAgreement(η) e **broadcastAskVotingRejection(η):** agente *med* solicita por votação a favor ou contra uma fórmula do argumento;

dtUpdate(DT, y, η , Agree, Reject, fac) e **dtUpdate(DT, y, is):** agente *med* atualiza a linha y na tabela de diálogos com o conjunto de agentes que apoiam e rejeitam uma fórmula, e também atualiza a força intrínseca do argumento;

askArguments(*ag*): agente *med* solicita por argumentos. O agente argumentador envia os argumentos em *S* e então *S* é esvaziada;

fillAgenda(agenda, listArgs) e **validateArguments(agenda):** agente *med* preenche a Agenda com os argumentos enviados pelo argumentador e valida estes argumentos, garantindo que não exista argumentos repetidos na tabela de diálogos e que os argumentos possuam algum tipo de vínculo uns com os outros (relações de ataque), além de não

aceitar argumentos com fórmulas que já foram apresentadas anteriormente em outros argumentos e estas fórmulas não foram consensualmente aceitas;

broadcastLearn(η , d , fac): agente *med* informa que a fórmula η ou $\neg\eta$ deve ser aprendida (ou ignorada caso $fac=0$). Os agentes argumentadores deverão atualizar a crença com seu rotulamento em K ou incorporar a nova crença com o rotulamento em KO .

3. Exemplo do Modelo

O modelo de diálogos proposto pode ser aplicado a diferentes tipos de problemas onde o consenso deve ser o ponto fundamental. A abordagem utilizada permite identificar o quanto um argumento deve ser aceito ou não pelo grupo de agentes e analisa cada informação contida na estrutura do argumento. O resultado do modelo é um conjunto de argumentos logicamente relacionados e um conjunto de fórmulas com rótulos que indicam o nível de consenso do grupo sobre elas.

Como exemplo, considere um grupo de três agentes apresentando suas opiniões sobre um filme e $\sigma = 0.5$. O framework pode ser definido como $FCC = \langle \{ag_1, ag_2, ag_3\}, \{0.4, 0.4, 0.2\}, med, \{d\}, 10 \rangle$ onde d representa o filme Piratas do Caribe. Considere os átomos das crenças representando as sentenças: a = piratas são legais, b = super-heróis, e = filme de luta, f = filme de ação. As crenças iniciais na base Σ dos agentes argumentadores são:

$$ag_1 = \{ \{a, a \rightarrow d, f, f \rightarrow \neg d, b \rightarrow d, e \rightarrow f\}, \{\} \}$$

$$ag_2 = \{ \{b, b \rightarrow \neg d, e, e \rightarrow f\}, \{\} \}$$

$$ag_3 = \{ \{a, a \rightarrow d, b, \neg e\}, \{\} \}$$

Inicialmente *med* cria o argumento inicial $\langle \{d\}, d \rangle$ e solicita por votos de apoio (linhas 6 a 12). Os agentes ag_1 e ag_3 apóiam esta informação, visto que possuem argumentos cujas conclusões são d . Esta fase de apoio poderá ser utilizada por algum mecanismo de tomada de decisões para atribuir uma força a esta alternativa (valor de preferência) caso não existam contra-argumentos. Os agentes ag_1 e ag_2 possuem contra-argumentos e se inscrevem para falar. Supondo que $wb = \{ag_1, ag_2\}$, o primeiro agente recebe o direito de fala, enviando o argumento $\langle \{f, f \rightarrow \neg d\}, \neg d \rangle$. O agente ag_2 apóia as fórmulas f e $\neg d$ com os argumentos $\langle \{e, e \rightarrow f\}, f \rangle$ e $\langle \{b, b \rightarrow \neg d\}, \neg d \rangle$. Com $fac(f)=0.8$ e $fac(\neg d)=0.8$, estas fórmulas devem ser consideradas como aceitas pelo grupo de agentes, enquanto que $fac(f \rightarrow \neg d)=0.4$ deve ser ignorada, não havendo consenso sobre ela. Todos os agentes devem atualizar suas bases de conhecimento com $ff[d(0.8)]$ e $\neg d[d(0.8)]$. O agente mediador insere todas as fórmulas com seus rótulos em *Bel*. O agente ag_3 solicita sua inscrição na fila para poder emitir contra-argumentos com $wb = \{ag_2, ag_3\}$.

O próximo agente da lista, ag_2 recebe o direito de fala, emitindo o argumento $\langle \{b, b \rightarrow \neg d\}, \neg d \rangle$ onde ag_3 apóia a fórmula b com $fac(b)=0.6$, e os agentes ag_1 e ag_3 apóiam $\neg d$ já que esta fórmula foi consensualmente aceita pelo grupo no argumento anterior. A fórmula $b[d(0.6)]$ é então atualizada no grupo de agentes. O agente ag_1 solicita sua inscrição na fila para poder emitir contra-argumentos com $wb = \{ag_3, ag_1\}$.

Como o agente ag_3 emitindo o argumento $\langle \{a, a \rightarrow d\}, d \rangle$, o agente ag_1 apóia todas as fórmulas, e os agentes ag_1 e ag_2 rejeitam d , visto que $\neg d$ é uma fórmula consensualmente aceita.

A Tabela 1 demonstra a tabela de diálogos para o assunto d . Durante a criação de argumentos, os agentes devem considerar apenas fórmulas que não possuam anotação para o assunto discutido, indicando que esta informação ainda não foi apresentada anteriormente, e fórmulas que apresentem valor de consenso diferente de zero. Argumentos contendo fórmulas que não são consensualmente aceitas devem ser ignorados. Esta validação também é realizada pelo agente mediador, não aceitando novos argumentos com fórmulas que não são aceitas pelo grupo, e também argumentos que já foram apresentados anteriormente. As crenças finais em Σ dos agentes argumentadores após diálogo são:

$$\begin{aligned}
 ag_1 &= \{ \{ a[d(0.6)], a \rightarrow d[d(0.6)], f[d(0.8)], f \rightarrow \neg d[d(0)], b \rightarrow d[d(0)], e \rightarrow f \}, \\
 &\quad \{ \neg d[d(0.8)], b[d(0.6)] \} \} \\
 ag_2 &= \{ \{ b[d(0.6)], b \rightarrow \neg d[d(0)], e, e \rightarrow f \}, \\
 &\quad \{ f[d(0.8)], \neg d[d(0.8)], a[d(0.6)], a \rightarrow d[d(0.6)] \} \} \\
 ag_3 &= \{ \{ a[d(0.6)], a \rightarrow d[d(0.6)], b[d(0.6)], \neg e \}, \\
 &\quad \{ f[d(0.8)], \neg d[d(0.8)] \} \}
 \end{aligned}$$

Tabela 1. Tabela de diálogos para o assunto d .

γ	ag	arg	$Attack$	$Support$	$Reject$	fac
1	med	$\langle \{d\}, d \rangle$	0	$d[ag_1, ag_3]$	$d[]$	$d=0.6$
2	ag_1	$\langle \{f, f \rightarrow \neg d\}, \neg d \rangle$	1	$f[ag_2]$ $f \rightarrow \neg d[]$ $\neg d[ag_2]$	$f[]$ $f \rightarrow \neg d[]$ $\neg d[]$	$f=0.8$ $f \rightarrow \neg d=0.4$ $\neg d=0.8$
3	ag_2	$\langle \{b, b \rightarrow \neg d\}, \neg d \rangle$	1	$b[ag_3]$ $b \rightarrow \neg d[]$ $\neg d[ag_1, ag_3]$	$b[]$ $b \rightarrow \neg d[]$ $\neg d[]$	$b=0.6$ $b \rightarrow \neg d=0.4$ $\neg d=1$
4	ag_3	$\langle \{a, a \rightarrow d\}, d \rangle$	2,3	$a[ag_1]$ $a \rightarrow d[ag_1]$ $d[ag_1]$	$a[]$ $a \rightarrow d[]$ $d[ag_1, ag_2]$	$a=0.6$ $a \rightarrow d=0.6$ $d=0.2$
5	ag_1	$\langle \{\neg d, b \rightarrow d\}, \neg b \rangle$	3	$\neg d[ag_2, ag_3]$ $b \rightarrow d[]$ $\neg b[ag_2]$	$\neg d[]$ $b \rightarrow d[ag_2]$ $\neg b[ag_2, ag_3]$	$\neg d=1$ $b \rightarrow d=0$ $\neg b=0.2$
6	ag_1	$\langle \{\neg d, a \rightarrow d\}, \neg a \rangle$	4	$\neg d[ag_2, ag_3]$ $a \rightarrow d[ag_2, ag_3]$ $\neg a[ag_2, ag_3]$	$\neg d[]$ $a \rightarrow d[]$ $\neg a[ag_2, ag_3]$	$\neg d=1$ $a \rightarrow d=1$ $\neg a=0.4$
7	ag_1	$\langle \{\neg d\}, \neg d \rangle$	4	$\neg d[ag_2, ag_3]$	$\neg d[]$	$\neg d=1$
8	ag_2	$\langle \{b\}, b \rangle$	5	$b[ag_1, ag_3]$	$b[]$	$b=1$

4. Características do Modelo

No modelo clássico para raciocínio sobre conhecimentos [Halpern, 1986; Fagin et al., 2004], o conceito de mundo possível representa os possíveis estados dos conhecimentos, permitindo que uma informação seja válida para um mundo possível e inválida para outros mundos. Seja $\mathcal{M} = \{m_1, \dots, m_m\}$ o conjunto de todos os mundos possíveis, ou seja, \mathcal{M} é um estado completo de mundos. O mundo possível $m_i \in \mathcal{M}$ representa um dos assuntos no mundo. O mundo atual é um mundo possível que está sendo discutido no momento do diálogo. No FCC, cada mundo possível representa uma alternativa de decisão, visto que uma informação pode ser aceita para uma alternativa e recusada para outra. Por exemplo: R\$25,00 pode ser considerado caro quando se trata do preço de um almoço em um restaurante de prato feito, enquanto que este mesmo valor pode ser considerado barato quando se trata de um almoço em um rodízio de pizzas.

Proposição 1: Um conjunto de todos os mundos possíveis \mathcal{M} pode ser expressado no FCC pelo conjunto de alternativas de decisão \mathcal{D} . Um mundo possível corresponde a cada

uma das alternativas de decisão $d_i \in \mathcal{D}$. O mundo atual corresponde na troca de argumentos entre os agentes em uma tabela de diálogos dt para uma alternativa de decisão, representado por dt_{d_i} .

Um conjunto de operadores modais são utilizados para representar as informações conhecidas pelos agentes, tais como K (conhecimento de um agente), E (conhecimento do grupo) e C (conhecimento comum). Para Fagin et al (2004), um agente conhece um fato η se η é necessariamente verdadeiro, ou seja, é verdadeiro em todos os mundos. Os operadores modais são usados para representar o conhecimento dos agentes. O operador $K_{ag_i}\eta$ indica que o agente ag_i conhece η , enquanto que $K_{ag_j}K_{ag_i}\eta$ indica que o agente ag_j sabe que ag_i conhece η . O operador modal $E_{AG}\eta$ indica que todos os agentes em AG conhecem o fato η , no entanto, cada agente não tem informação se os demais agentes conhecem ou não η . O conhecimento comum é definido como o “estado do conhecimento onde todos sabem, todos sabem que todos sabem, todos sabem que todos sabem que todos sabem, etc.”. Neste sentido, o operador modal $C_{AG}\eta$ indica que η é conhecido por todos os agentes e todos os agentes sabem que os demais agentes também conhecem η .

Proposição 2: Tem-se que $K_{ag_i}\eta$ é equivalente a $\eta \in \Sigma_i$. Um agente pode se comunicar diretamente com outros agentes na busca por informações. Temos que em $K_{ag_j}K_{ag_i}\eta$, o agente ag_j se comunicou com ag_i sobre η e obteve uma resposta. É possível representar informações desconhecidas, tal como $\neg K_{ag_j}K_{ag_i}\eta$ (o agente ag_j não sabe que ag_i conhece η) quando não existe comunicação entre os agentes ou não existe resposta nas comunicações.

Proposição 3: Tem-se que $E_{AG}\eta$ é equivalente a $\forall ag_i \in AG: K_{ag_i}\eta$.

Proposição 4: Todas as fórmulas consensualmente aceitas são atualizadas nas bases de conhecimento dos agentes. Estas fórmulas recebem rótulos indicando a alternativa de decisão e o valor de consenso. Desta forma, $C_{AG}\eta \Leftrightarrow \exists \eta[d_j(fac)] \in \Sigma_i$ com $d_j \in \mathcal{D}$, ou seja, se existe uma fórmula contendo rótulos na base de conhecimento de um agente, esta fórmula já foi apresentada anteriormente e aceita consensualmente pelo grupo, tornando-se conhecimento comum.

Exemplo 2: Em uma base de conhecimentos $\Sigma_i = \{x, y[d(0.5)], z, \neg z[d(0.5)], a[d(0.6)]\}$, o agente ag_i considera x como aceito para todos os mundos possíveis; $y[d(0.5)]$ é também aceito para todos os mundos possíveis; z é aceito para todos os mundos, com exceção do mundo d que possui $fac(\neg z) = 0.5$; e, $a[d(0.6)]$ é verdadeiro para todos os mundos. O modelo também representa informações que são conhecimento comum em um mundo possível no grupo de agentes, tais como $y, \neg z$ e a .

Todas as fórmulas em Σ_i que não possuem anotações são consideradas aceitas para todos os mundos possíveis pelo agente ag_i . Quando uma fórmula recebe um rótulo, este rótulo contém uma anotação para um mundo possível e um valor de consenso. Este rótulo indica que a fórmula foi consensualmente aceita pelo grupo de agentes para o mundo anotado e que esta informação é um conhecimento comum. Quando a base de conhecimentos de um agente possui fórmulas contraditórias (ex.: η e $\neg\eta$), o agente utiliza a fórmula como segue:

1. Inicialmente considera a fórmula que possui o mundo atual anotada no rótulo;
2. Caso o mundo atual não esteja anotado em nenhum dos rótulos, o agente utiliza:

- a) A fórmula da sua base K_i (conhecimento do agente aplicado a todos os mundos que não estejam anotados);
- b) A fórmula da sua base KO_i que se aplica ao maior número de domínios (conhecimento do agente adquirido durante o diálogo);
- c) A fórmula da sua base KO_i que apresente o maior valor de consenso, ou;
- d) Escolha aleatória da fórmula.

Para que a tomada de decisões seja colaborativa e consensual, não devem existir hierarquias entre os agentes, todas as alternativas de decisão devem ser debatidas e todas as informações que os agentes possuem sobre a discussão devem ser consideradas. Quando $ts_1 = \dots = ts_n$, todos os agentes possuem o mesmo valor de *expertise*, indicando que não existe um agente autoritário ou mais influenciador no diálogo. No *FCC*, embora exista a presença de um mediador, este agente não atua como tomador de decisões. O agente *med* é utilizado para a coordenação da comunicação e sincronização entre os agentes durante o envio dos argumentos. Além disso, o mediador também realiza uma validação nos argumentos recebidos dos agentes argumentadores. Esta validação se torna necessária para que um argumento sempre esteja vinculado a outros argumentos no diálogo, não hajam argumentos repetidos no diálogo (gerando *loop* na argumentação) ou argumentos que possuem informações já apresentadas anteriormente para o mundo atual e que não foram amplamente aceitas pelo grupo.

5. Conclusões e Trabalhos Futuros

Neste trabalho foi apresentado um modelo para formação do consenso em diálogos argumentativos aplicados a sistemas multiagentes. Como resultado, cada agente argumentativo do sistema possui sua base de conhecimentos contendo crenças representando fatos e regras, é dotado da capacidade de construir argumentos a favor ou contra determinada alternativa de decisão (ou assunto em discussão), e pode atualizar sua base de conhecimentos com crenças que são apoiadas pela maioria dos agentes argumentadores. Para determinar quais crenças são apoiadas ou rejeitadas pelo grupo, utiliza-se a *expertise* dos agentes e um valor limiar que indica se a quantia de apoios é suficiente para considerar uma crença aceita pelo grupo.

Ao final de um diálogo sobre um assunto, os agentes são capazes de saber quais são as crenças tidas como conhecimento comum, ou seja, crenças que são aceitas por consenso pela maioria dos agentes no grupo. Também foram propostas algumas regras para que os agentes argumentadores decidam qual a crença será utilizada ao criar argumentos e votar para apoio/rejeição quando solicitado.

Um agente mediador foi proposto com o intuito de conduzir o diálogo, sincronizando a troca de argumentos entre os agentes, e validando estes argumentos. Fica a cargo do agente mediador as requisições por argumentos, requisições por votos, cálculo da função de atualização de crenças e as requisições para que os agentes argumentadores atualizem suas bases de conhecimento.

O framework para a construção do conhecimento comum pode ser estendido de diversas formas. Para trabalhos futuros, propomos algumas direções: (1) tendo-se a tabela de diálogos para cada assunto discutido, esta tabela pode ser mapeada para um grafo de argumentos; (2) cada argumento possui um conjunto de fórmulas e cada fórmula possui um valor de consenso obtido pela função de atualização de crenças. O conjunto de valores de consenso de um argumento pode ser utilizado para o cálculo da força consensual do

argumento. Utilizando as relações de ataques no grafo de argumentos, pode-se elaborar um modelo para o cálculo da força global dos argumentos e a aplicação de semânticas em um framework de argumentação ponderado (ex.: [Da Costa Pereira et al., 2011]; [Leite e Martins, 2010]); (3) elaboração de uma semântica para a tomada de decisões consensuais que considere os diálogos para cada alternativa de decisão, o conjunto de argumentos fornecidos e a força consensual destes argumentos.

Por fim, acreditamos que o framework para a construção do conhecimento comum pode ser aplicado em qualquer domínio onde seja necessário a identificação do consenso de um grupo de agentes sobre determinada fórmula ou argumento. Um sistema baseado neste framework consegue distinguir informações entre mundos possíveis e manter a concisão com os operadores modais aplicados aos sistemas de mundos possíveis.

Referências

- Amgoud, L., and Prade, H. (2009). "Using arguments for making and explaining decisions". *Artificial Intelligence*, pages 413-436.
- Amgoud, L., Belabbes, S., and Prade, H. (2005). "Towards a formal framework for the search of a consensus between autonomous agents". In *International Workshop on Argumentation in Multi-Agent Systems*, pages 264-278.
- Besnard, P., and Hunter, A. (2014). "Constructing argument graphs with deductive arguments: a tutorial". *Argument & Computation*, pages 5-30.
- Da Costa Pereira, C., Tettamanzi, A. G., and Villata, S. (2011). "Changing one's mind: Erase or rewind? possibilistic belief revision with fuzzy argumentation based on trust". In: *International Joint Conference on Artificial Intelligence*, pages 164-171.
- Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. (2004). "Reasoning about knowledge". *MIT press*.
- Grossi, D., and Pigozzi, G. (2014). "Judgment aggregation: a primer". *Synthesis Lectures on Artificial Intelligence and Machine Learning*, pages 1-151.
- Halpern, J. Y. (1986). "Reasoning about knowledge: an overview". In: *Conference on Theoretical aspects of reasoning about knowledge*, pages 1-17.
- Hertzberg, M., Smith, R., and Westphal, R. A. (2013). "Consensus Handbook: Cooperative decision-making for activists, co-ops and communities". *Seeds for Change Lancaster Co-operative Ltd*.
- Leite, J., and Martins, J. (2011). "Social abstract argumentation." In: *Twenty-Second International Joint Conference on Artificial Intelligence*, vol. 11, pages 2287-2292.
- Parsons, S., and McBurney, P. (2003). "Argumentation-based dialogues for agent coordination". *Group Decision and Negotiation*, pages 415-439.
- Romme, A. G. L. (2004). "Unanimity rule and organizational decision making: A simulation model". *Organization Science*, pages 704-718.
- Zhong, Q., Fan, X., Toni, F., and Luo, X. (2014). "Explaining Best Decisions via Argumentation". In *European Conference on Social Intelligence*, pages 224-237.

Uma abordagem baseada em *Simulated Annealing* aplicada ao problema de programação de jogos do Campeonato Mineiro de Futebol

Keline M. Balieiro¹, Laércio I. Santos¹, Luciana B. Cosme¹, Renato A. C. Silva¹

¹Instituto Federal do Norte de Minas Gerais (IFNMG) *Campus Montes Claros*
Montes Claros – MG – Brasil

keline.balieiro@gmail.com, laercio.santos@ifnmg.edu.br

luciana.cosme@ifnmg.edu.br, renato.cota@ifnmg.edu.br

Abstract. *The generation of complete set of rounds in sports tournaments is an optimization problem that takes into account the characteristics of each competition. This work deals with the timetabling related to the Football Championship of Minas Gerais State in Brazil. The goal of the proposal is to minimize the difference between the largest and smallest distances traveled by the teams during the tournament. To solve this problem, it is proposed an approach based on Simulated Annealing and Polygon Method. The evaluation of the results is done using the 2016 and 2017 real instances and the proposed schedules show that the differences between distances are effectively minimized, if compared to the actual timetables.*

Resumo. *A Programação de Jogos em torneios esportivos é um problema de otimização que leva em consideração as características de cada competição. Este trabalho aborda a programação de jogos referentes ao Campeonato Mineiro de Futebol e tem como objetivo a minimização da diferença entre a maior e a menor distâncias percorridas durante o torneio. Propõe-se uma abordagem baseada nas técnicas Simulated Annealing e Método do Polígono. A avaliação dos resultados é feita utilizando as instâncias reais de 2016 e 2017 e mostram que as diferenças entre as distâncias são efetivamente minimizadas, se comparadas com a tabela real de jogos.*

1. Introdução

A confecção da programação de jogos em torneios esportivos é uma classe importante de problemas combinatórios. Entretanto, o tratamento desse problema é uma tarefa difícil e de grande importância, pois as competições esportivas representam uma das maiores atividades econômicas do mundo, especialmente no Brasil, pela sua significância cultural e financeira. De acordo com [Segundo et al. 2015], o futebol se enquadra como a modalidade esportiva mais popular, estando presente na maioria dos países. Segundo [Capelo 2017], o faturamento dos principais times brasileiros no ano de 2016 aumentou em torno de 41% em relação ao ano de 2015, que foi equivalente a R\$ 5 bilhões. Apesar do aumento significativo da arrecadação, o valor da dívida das 24 principais equipes brasileiras (vinte das equipes que jogaram a primeira divisão do Campeonato Brasileiro em

2016 e as quatro que estavam na segunda divisão em 2016 e estão disputando a primeira em 2017) teve um aumento de 2%, chegando em torno de R\$ 6,4 bilhões. A presença dessa dívida, apesar do aumento da receita, mostra que os clubes ainda precisam equilibrar seus orçamentos, seja pelo aumento das receitas seja pelo uso racional e otimizado dos recursos disponíveis.

Em relação ao problema, é importante ressaltar que os jogos são disputados entre dois times e todos os participantes jogam em todas as rodadas com oponentes diferentes. As partidas são realizadas em vários locais ao longo de um determinado período de tempo. Assim, faz-se necessário elaborar um bom escalonamento dos jogos, para que as escalas geradas sejam menos dispendiosas e atendam às restrições definidas pela entidade organizadora.

O problema aborda vários tipos de eventos esportivos pelo mundo. Cada competição possui restrições e particularidades específicas, que devem ser tratadas. Alguns trabalhos, como [Della Croce and Oliveri 2006] e [Kostuk and Willoughby 2012] tratam o problema para campeonatos Italiano e Canadense de futebol, respectivamente. Os autores [Biajoli et al. 2004], [Biajoli and Lorena 2005], [Souza et al. 2005], [Biajoli and Lorena 2006], [Ribeiro and Urrutia 2007] e [Ribeiro and Urrutia 2012] abordam o problema para eventos de futebol, referentes ao Campeonato Brasileiro da série A. Outros esportes também são explorados através desse tipo de problema, como o Basebol, com [Anagnostopoulos et al. 2006] e também o Basquete [Hoshino and Kawarabayashi 2011]. É possível verificar que não há enfoque para as competições regionais brasileiras. Essas competições normalmente envolvem times menores que possuem orçamento mais restrito e, por isso, mais dependentes de um bom uso dele, até mesmo para o bom desempenho e permanência no futebol.

Na literatura pesquisada para o problema, é comum encontrar trabalhos que usam Algoritmos Genéticos (AGs) com representação compacta dos jogos e times, [Von Zuben et al. 2002] e com o tratamento das soluções factíveis. Há ainda alguns trabalhos que usam AGs, *Simulated Annealing* e Método do Polígono (PL) [Biajoli and Lorena 2005] e [Biajoli and Lorena 2006], e alguns outros trabalhos que usam SA com a representação expandida dos jogos e times [Biajoli et al. 2003], [Anagnostopoulos et al. 2006], [Lim et al. 2006], [Menon and Jha 2013]. Além disso, normalmente os trabalhos que abordam o problema têm como objetivo principal a minimização da distância total percorrida pelas equipes no decorrer do campeonato. Para este trabalho, minimizar a distância total não é interessante, visto que a maioria dos jogos, do campeonato regional que será estudado, são realizados aos finais de semana, o que torna inviável que os times se desloquem do destino anterior diretamente para o próximo destino, como amplamente utilizado na literatura. Também há o problema que as tabelas reais confeccionadas podem não ser justas para todos os times. Por exemplo, na edição de 2016, o time de Uberlândia Futebol Clube percorreu 7628 quilômetros, enquanto o América Futebol Clube percorreu 2408 quilômetros. A diferença entre as distâncias foi de 5520 quilômetros, o que significa que o Uberlândia percorreu uma distância de aproximadamente três vezes superior ao América. Diante disso, e na busca por uma igualdade no deslocamento, este trabalho propõe a confecção da tabela do Campeonato Mineiro de Futebol utilizando uma nova abordagem com base na metaheurística *Simulated Annealing* e na heurística de expansão de código, o Método do Polígono, com o objetivo de minimizar

a diferença entre a maior e a menor distâncias percorridas durante a competição, aplicada ao Campeonato Mineiro de Futebol. As principais diferenças para os trabalhos citados são: 1) a proposta utiliza o método SA, na sua forma tradicional, que é mais simples de implementar que os AGs; 2) a modelagem proposta trata com maior flexibilidade as restrições e 3) este trabalho utiliza uma representação compacta ao invés da representação expandida, normalmente encontrada nos trabalhos que utilizam SA.

O trabalho está organizado da seguinte maneira: a seção 2 apresenta a definição do problema, a seção 3 aborda o método proposto para resolução do problema, na seção 4 os resultados são apresentados e discutidos e, por fim, na seção 5 são apresentadas as considerações finais.

2. Definição do Problema

O Problema de Programação de Jogos (PPJ) foi proposto inicialmente por [Easton et al. 2001], sendo uma abstração do problema real da Major League Baseball (MLB) dos Estados Unidos. O problema consiste em elaborar uma solução em que ocorram disputas dois a dois entre os participantes e que todos os times joguem em todas as rodadas com oponentes diferentes na sua sede ou fora dela. Assim como proposto por Easton, normalmente os trabalhos sobre o tema tem como objetivo principal a minimização do somatório de todas as distâncias percorridas pelos times no decorrer da competição, visando a redução do custo com o transporte dos participantes. Essa abordagem só é possível de ser aplicada se o time pode se deslocar de um destino diretamente para o próximo. Entretanto, existem estudos mais recentes em que outros objetivos são propostos, como a redução da maior distância percorrida por um dos times, de forma a promover maior igualdade de deslocamento entre os concorrentes. A complexidade do problema pode ser percebida, por exemplo, em uma competição envolvendo n times, confrontando-se entre si em turnos completos (onde todos os times jogam em cada rodada), o número de combinações possíveis é dado pela Eq. 1 [Von Zuben et al. 2002]:

$$(n-1)!(n-3)!(n-5)! \dots (n-(n-1))! \times 2^{(n-1) \times n/2} \quad (1)$$

onde n é o número de times participantes. Em uma competição com 20 times participantes haverá $2,90 \times 10^{130}$ combinações possíveis.

O PPJ pode ser aplicado na primeira fase do Campeonato Mineiro de Futebol, levando em consideração as seguintes restrições:

- Todos os clubes se enfrentarão uma vez, em turno único, classificando-se ao final das 11 rodadas, para a fase semifinal, os quatro melhores colocados;
- São clássicos os jogos entre América, Atlético e Cruzeiro, na primeira fase, sendo que cada um dos três clubes exercerá seu mando de campo em um clássico;
- A diferença da quantidade de jogos disputados em casa ou fora de casa é de um jogo;
- Cruzeiro e Atlético não jogam em casa na mesma rodada;
- Os times poderão jogar em casa ou fora por até duas rodadas seguidas;
- Pelo menos um, entre Cruzeiro e Atlético, jogam, necessariamente em casa em cada rodada;
- Cada equipe joga uma vez com todas as outras equipes, em turno único, em sua sede ou fora dela.

2.1. Formulação do Problema

Uma programação de jogos esportivos pode ser modelada como um problema de otimização, em que cada possível jogo entre um par de times representa uma variável de decisão. Pode-se formular a confecção da tabela de jogos esportivos como um problema da seguinte forma:

$$f(x) = \max(\text{custo}(x)) - \min(\text{custo}(x)) + \sum w_i * \text{pen}_i \quad (2)$$

Na Eq. 2, x é a programação dos jogos representada por uma matriz de variáveis de decisão composta por T times e R rodadas; a função $\text{custo}(x)$ retorna um vetor em que cada posição representa a distância percorrida por cada time de T em todas as rodadas R , presentes da matriz x ; $\text{pen}_i, i = \{1, 2, 3, 4\}$, indica quantas vezes a restrição i foi violada e w_i é o peso atribuído a i ésima restrição;

Neste trabalho, consideraram-se as seguintes restrições:

1. os times devem jogar em casa ou fora até duas rodadas seguidas;
2. para os jogos entre América, Atlético e Cruzeiro, na primeira fase, cada um dos três clubes deve exercer um mando de campo;
3. Cruzeiro e Atlético não devem jogar em casa na mesma rodada;
4. pelo menos um, entre Cruzeiro e Atlético, jogam necessariamente em casa em cada rodada.

A Tabela 1 representa a programação dos jogos (x) de um campeonato fictício com os mandos de campo. Essa tabela é lida da seguinte maneira: as linhas representam os jogos dos times, com $i = 1 \dots T$, as colunas representam as rodadas, com $j = 1 \dots R$. Cada célula da rodada possui o número do oponente do time da linha i naquela rodada j . Quando o sinal negativo (-) aparece antes do número do time adversário significa que o time i joga fora de casa, portanto, a distância percorrida pelo time adversário é contabilizada. Por exemplo, na primeira rodada temos os jogos Cruzeiro (1) x Atlético(2), sendo que o mando de campo é do Atlético, e América (3) x Vila Nova (4), sendo que o mando de campo é do América.

Tabela 1. Programação de jogos com os mandos de campo

Time/Rodada	1	2	3
Cruzeiro (1)	-2	+3	-4
Atlético (2)	+1	-4	-3
América (3)	+4	-1	+2
Vila Nova (4)	-3	+2	+1

A formulação proposta transforma as restrições em penalidades, pois dessa forma não é preciso tratar no algoritmo todas as soluções não factíveis. [Biajoli and Lorena 2006] mostram que o desenvolvimento de operadores para manter a factibilidade não é uma tarefa fácil e tem custo computacional elevado. Além disso, a tabela real de jogos em 2017 fere uma restrição enquanto a tabela de 2016 não. Portanto, essa formulação permite maior flexibilidade de como tratar cada restrição, como por exemplo, permitir alguma solução não factível ou a exclusão e a inclusão de novas restrições com maior facilidade.

Para resolver a Eq. 2, é necessário determinar, dentre todos os pontos possíveis, o ponto x^* em que a função objetivo retorna o menor valor, o que dizer que, $f(x^*)$ é mínimo. Os detalhes da resolução do problema serão apresentados na seção 3.

3. Abordagem proposta

Uma forma de resolver o problema é utilizar técnicas determinísticas, tal como o método *Branch and Price* [Barnhart et al. 1998] que, apesar de garantir a obtenção do ótimo global, possui alto custo computacional. Em [Easton et al. 2003], os autores utilizaram essa técnica, porém, só conseguiram resolver uma tabela com 8 times, que levou 4 dias para ser obtida. Devido às dificuldades mencionadas, os métodos metaheurísticos são bastante utilizados. Para o presente trabalho, foi utilizada a metaheurística *Simulated Annealing* (SA) [Kirkpatrick et al. 1983] juntamente com o Método do Polígono [Biajoli and Lorena 2006] devido especialmente à maior simplicidade de implementação e dos bons resultados encontrados.

3.1. Método do Polígono

Uma representação compacta dos indivíduos é utilizada para a aplicação da metaheurística *Simulated Annealing*, em que cada indivíduo representa um time. Então, o Método do Polígono [Dinitz et al. 1995] é submetido e associado à metaheurística. O vetor de representação de indivíduos é inicialmente gerado de maneira aleatória.

Tabela 2. Exemplo de uma representação de indivíduos para um torneio com 4 times apresentada em [Biajoli and Lorena 2006]

3	2	5	1	6	4
---	---	---	---	---	---

A execução do Método do Polígono se inicia quando há a escolha de um time base, que é escolhido de maneira aleatória. O time que for selecionado é inserido na primeira posição do vetor V . Os outros times são arranjados nas posições restantes do vetor (da posição 2 até T). Em cada rodada ($R=1, \dots, T-1$), o time base confronta com o time da posição $i=2$, definindo, assim, o jogo $T1, T2$. Os times das posições $i=3, \dots, (T/2)+1$ jogam com os times das posições $j=T-i+3$, definindo, assim, os demais jogos. Após a definição dos jogos da rodada R , os times que estão dispostos nas posições $i=2, \dots, T$ são rotacionados no sentido horário do vetor, de forma que: os times que estão nas posições $i=3, \dots, T$ passam para a posição $i-1$ e o time que estiver na posição $i=2$ vai para a posição n . Dessa forma, o vetor mostrado na Tabela 2 se transforma na matriz mostrada na Tabela 1, sem os mandos de campo. Esses mandos de campo são gerados a partir da heurística proposta em [Biajoli and Lorena 2006].

A Figura 1 apresenta um exemplo da execução do Método do Polígono para um torneio com 6 times e 5 rodadas. Nesse exemplo, o time 3 é escolhido, aleatoriamente, para ser o time base e, na rodada 1, os demais times, 2, 5, 1, 6, 4, nessa sequência, são escolhidos também de forma aleatória. Assim, na primeira rodada, temos: o time 3 joga com o time 2, o time 5 joga com o time 1 e o time 6 joga com o time 4. Para a rodada 2, acontece a rotação dos times, de modo que, o time 2 vai para a ultima coluna e os demais times voltam uma posição, cada. Assim, temos a rodada 2 com time 3 jogando com o time 5, time 1 com o time 6 e time 4 com o time 2. Na rodada 3 o time 5, que está

na segunda coluna, vai para a última coluna e o processo continua da mesma forma, até chegar à rodada 5.

Rodada 1	3	2	5	1	6	4
Rodada 2	3	5	1	6	4	2
Rodada 3	3	1	6	4	2	5
Rodada 4	3	6	4	2	5	1
Rodada 5	3	4	2	5	1	6

Figura 1. Método do Polígono. Fonte: [Biajoli and Lorena 2005]

3.2. Simulated Annealing

O *Simulated Annealing* (SA) ou Recozimento Simulado é um método de busca local que tem como objetivo encontrar a melhor solução entre um número finito de soluções existentes para um determinado problema. Em [Kirkpatrick et al. 1983] é apresentado o primeiro algoritmo para resolver problemas de otimização baseado no modelo original do SA. Esse método possui a capacidade de fugir de ótimos locais. Isso ocorre devido ao fato de ele aceitar eventualmente uma solução pior do que a solução atual. No início da execução do algoritmo, existem altas probabilidades de aceitar uma solução pior, ou seja, o método tenta varrer grande parte do espaço de soluções. Ao final da execução do algoritmo, a probabilidade de aceitar uma solução pior é menor em relação ao início. Essa técnica é utilizada com objetivo de refinar a busca. O SA realiza o processo de otimização simulando os níveis de temperatura no resfriamento. Essa abordagem permite que a vizinhança de uma solução seja explorada durante a execução do algoritmo.

Para ilustrar os detalhes do algoritmo proposto, sumarizado no Algoritmo 1, considere que o valor de s representa o estado, vetor de times, na iteração atual, como mostrado na Tabela 2, s^* representa o melhor estado alcançado até a iteração atual, a função $Poligono(s)$ recebe uma solução s e retorna uma matriz x com todos os jogos e mandos de campo, a função $CalculaFuncaoCusto(x)$ calcula o valor da função objetivo para x , apresentada na Eq. 2, $randi$ é uma função que retorna um inteiro uniformemente distribuído no intervalo de $1..T$. Serão, portanto, selecionados aleatoriamente dois inteiros $p1$ e $p2$, que representam as posições a serem trocadas no vetor s na função $TrocaPosicao$. Por fim, T e L são parâmetros do SA a serem configurados pelo usuário.

O algoritmo SA utiliza pouco espaço de memória se comparado a métodos populacionais como Algoritmos Genéticos. Isso porque não é necessário armazenar um conjunto de soluções possíveis para o problema tratado, e sim uma única solução possível (tabela de jogos). Além disso, os operadores do SA são mais simples de serem aplicados que operadores de métodos baseados em população.

4. Resultados e Discussão

Para avaliar o desempenho da abordagem proposta, foram utilizadas as instâncias de 2016 e 2017 do Campeonato Mineiro de Futebol. Outras instâncias não foram utilizadas porque os regulamentos utilizados são diferentes. O objetivo do trabalho é, então, encontrar tabelas de jogos mais otimizadas que a tabela real, no que diz respeito à diferença entre a

Algoritmo 1: Simulated Annealing Proposto

```

1  início
2  |   s=solução Inicial, s*=s, T=T0;
3  |   x=Poligono(s);
4  |   f(x)= CalculaFuncaoCusto(x);
5  |   f(x*)=f(x);
6  |   while T > 0 do
7  |       for i=1 : L do
8  |           p1=randi(1...T);
9  |           p2=randi(1...T);
10 |           s'=TrocaPosicao(s, p1, p2);
11 |           x'=Poligono(s');
12 |           f(x')= CalculaFuncaoCusto(x');
13 |           Δf=f(x') - f(x);
14 |           if Δf < 0 then
15 |               s=s';
16 |               f(x)=f(x');
17 |               if f(x') < f(x*) then
18 |                   s*=s';
19 |                   f(x)=f(x');
20 |               end
21 |           end
22 |           else if uniforme(0, 1) < exp(-Δf/T) then
23 |               s=s';
24 |           end
25 |       end
26 |       T=T - 1;
27 |   end
28 fim

```

maior e a menor distância percorrida, utilizando os métodos SA e Polígono. Além disso, os parâmetros de ajuste do SA são $T_0 = 1000$, $L = 100$.

4.1. Tabelas de Distâncias

As tabelas de distâncias¹ são utilizadas para calcular a distância percorrida por cada time na tabela de times e rodadas. As Tabelas 3 e 4 apresentam os valores correspondentes às distâncias entre as cidades sedes do times, nos anos de 2016 e 2017, respectivamente. Na Tabela 3, por exemplo, significa que a distância entre o time AME-BH(1)² (na linha 1) e

¹As distâncias foram calculadas com o auxílio da ferramenta Google Maps, levando em consideração o menor trajeto entre as cidades.

²Por questões de falta de espaço nas tabelas, os nomes dos times foram abreviados, da seguinte forma: América Futebol Clube (AME-BH), América Futebol Clube(AME-TO), Associação Atlética Caldense(CAL), Boa Esporte Clube(BOA), Clube Atlético Mineiro(ATL), Clube Atlético Tricordiano (TRI), Cruzeiro Esporte Clube(CRU), Esporte Clube Democrata(DEM), Guarani Esporte Clube(GUA), Tombense Futebol Clube(TOM), Uberlândia Esporte Clube(UBE), União Recreativa dos Trabalhadores(URT), Villa Nova Atlético Clube(VIL).

o time CAL (2) (na coluna 2) é de 448 quilômetros e entre o time AME-BH (1) e o time BOA (3) (na coluna 3) é 313 quilômetros, porém, quando a distância é zero significa que os times são da mesma cidade.

Tabela 3. Tabela de Distâncias - 2016

Time/Time	1	2	3	4	5	6	7	8	9	10	11	12
AME-BH(1)	0	448	313	0	293	0	122	368	273	538	399	22
CAL(2)	448	0	153	448	183	448	359	660	449	470	560	465
BOA(3)	313	153	0	313	35	313	242	531	323	559	444	329
ATL(4)	0	448	313	0	293	0	122	368	273	538	399	22
TRI(5)	293	183	35	293	0	293	220	497	280	588	450	310
CRU(6)	0	448	313	0	293	0	122	368	273	538	399	22
GUA(7)	122	359	242	122	220	122	0	479	335	458	319	138
TOM(8)	368	660	531	368	497	368	479	0	219	909	771	377
TUP(9)	273	449	323	273	280	273	33	219	0	800	662	283
UBE(10)	538	470	559	538	588	538	458	909	800	0	224	555
URT(11)	399	560	444	399	450	399	319	771	662	224	0	416
VIL(12)	22	465	329	22	310	22	138	377	283	555	416	0

Tabela 4. Tabela de Distâncias - 2017

Time/Time	1	2	3	4	5	6	7	8	9	10	11	12
AME-BH(1)	0	447	448	0	293	0	317	368	273	538	399	22
AME-TO(2)	447	0	905	447	737	447	138	422	588	981	822	479
CAL(3)	448	905	0	448	183	448	775	660	449	470	560	465
ATL(4)	0	447	448	0	293	0	317	368	273	538	399	22
TRI(5)	293	737	183	293	0	293	607	497	280	588	450	310
CRU(6)	0	447	448	0	293	0	317	368	273	538	399	22
DEM(7)	317	138	775	317	607	317	0	289	455	851	697	348
TOM(8)	368	422	660	368	497	368	289	0	219	909	771	377
TUP(9)	273	588	449	273	280	273	455	219	0	800	662	283
UBE(10)	538	981	470	538	588	538	851	909	800	0	224	555
URT(11)	399	822	560	399	450	399	697	771	662	224	0	416
VIL(12)	22	479	465	22	310	22	348	377	283	555	416	0

4.2. Soluções encontradas

As Tabelas 5 e 6 mostram as matrizes de jogos (solução) encontradas pelo algoritmo proposto, para os anos de 2016 e 2017, respectivamente. Nessas tabelas, as linhas representam os times, as colunas representam as rodadas, o valor representa o adversário, e o sinal o mando de campo. Por exemplo, pela Tabela 5 o América-MG jogou contra o Cruzeiro na segunda rodada, na sede do Cruzeiro.

A Tabela 7 apresenta os critérios utilizados na comparação da presente proposta com a tabela real disponibilizada pela Federação Mineira de Futebol(FMF), sendo eles, a máxima e a mínima distância e a diferença entre elas.

Tabela 5. Tabela com a programação de jogos e mandos de campo gerada pelo algoritmo proposto - 2016

Time/Rodada	1	2	3	4	5	6	7	8	9	10	11
AME-BH(1)	11	-6	8	-10	7	-3	-2	4	-9	5	-12
CAL(2)	10	-5	-7	12	-3	-11	1	-6	4	-8	9
BOA(3)	6	-8	10	-7	2	1	-4	9	-5	12	-11
ATL(4)	12	-11	6	-8	10	-7	3	-1	-2	9	-5
TRI(5)	9	2	-12	-11	6	-8	10	-7	3	-1	4
CRU(6)	-3	1	-4	9	-5	12	-11	2	8	-10	7
GUA(7)	-8	-10	2	3	-1	4	-9	5	-12	11	-6
TOM(8)	7	3	-1	4	-9	5	-12	11	-6	2	-10
TUP(9)	-5	12	11	-6	8	-10	7	-3	1	-4	-2
UBE(10)	-2	7	-3	1	-4	9	-5	12	-11	6	8
URT(11)	-1	4	-9	5	-12	2	6	-8	10	-7	3
VIL(12)	-4	-9	5	-2	11	-6	8	-10	7	-3	1

Tabela 6. Tabela com a programação de jogos e mandos de campo gerada pelo algoritmo proposto - 2017

Time/Rodada	1	2	3	4	5	6	7	8	9	10	11
AME-BH(1)	11	-4	2	-10	5	-3	-9	6	-8	7	-12
AME-TO(2)	5	3	-1	6	-8	7	-12	11	-4	9	-10
CAL(3)	4	-2	10	-5	9	1	-6	8	-7	12	-11
ATL(4)	-3	1	-6	8	-7	12	-11	9	2	-10	5
TRI(5)	-2	-10	9	3	-1	6	-8	7	-12	11	-4
CRU(6)	12	-11	4	-2	10	-5	3	-1	-9	8	-7
DEM(7)	8	9	-12	-11	4	-2	10	-5	3	-1	6
TOM(8)	-7	12	11	-4	2	-10	5	-3	1	-6	-9
TUP(9)	10	-7	-5	12	-3	-11	1	-4	6	-2	8
UBE(10)	-9	5	-3	1	-6	8	-7	12	-11	4	2
URT(11)	-1	6	-8	7	-12	9	4	-2	10	-5	3
VIL(12)	-6	-8	7	-9	11	-4	2	-10	5	-3	1

Tabela 7. Critérios de Comparação

Instância	Tabela Real			Algoritmo Proposto		
	Máximo	Mínimo	Diferença	Máximo	Mínimo	Diferença
2016	7628	2408	5220	5134	3086	2048
2017	6768	2546	4222	5766	3298	2468

Pela Tabela 7, é possível perceber que na edição do Campeonato Mineiro de Futebol de 2016, a equipe que percorreu a maior distância durante a competição, com o total de 7628 quilômetros, foi o Uberlândia Futebol Clube (Uberlândia). Enquanto a equipe que percorreu a menor distância, 2408 quilômetros, foi o América Futebol Clube (Belo Horizonte). A diferença entre as distâncias foi de 5220 quilômetros. O algoritmo proposto gerou uma solução em que a diferença das distâncias foi de 2048 quilômetros, isto significa que a diferença foi minimizada em 60,77%, se comparado à tabela real. Já, na

tabela gerada, o time que mais percorreu foi a União Recreativa dos Trabalhadores (Patos de Minas), com 5134 quilômetros e o Cruzeiro Esporte Clube (Belo Horizonte) percorreu 3086 quilômetros, sendo uma distância bem menor que na tabela real.

Para a edição de 2017, o Esporte Clube Democrata (Governador Valadares) percorreu a maior distância, correspondente a 6768 quilômetros. O Villa Nova teve a menor distância, com 2546 quilômetros. Os resultados obtidos com o algoritmo proposto foram 5766 quilômetros para a maior distância, pertencente à equipe Uberlândia Futebol Clube (Uberlândia), e a menor distância, pertencente à equipe América Futebol Clube (Belo Horizonte), foi de 3298 quilômetros. Portanto, houve uma redução de 41,54%, se comparado com o resultado alcançado com a tabela real. A diferença entre a tabela real e a encontrada foi 1754 quilômetros. Isso mostra que as tabelas reais de 2016 e 2017 poderiam ter uma redução significativa dos custos gerais de transporte para alguns times, especialmente para aqueles longe da capital, sem prejudicar os demais.

Na Tabela 8, é possível observar a distância percorrida por cada time na tabela real e na tabela encontrada, em cada instância. E ainda, uma projeção da distância total percorrida por cada time se todos os seus jogos fossem realizados fora de casa. Ou seja, mostra quais times estão em regiões mais centrais ou distantes. Pode-se perceber que os times América-BH Atlético, Cruzeiro e Villa Nova são os times mais centrais em ambas as edições, e que, Tombense, Uberlândia e URT na edição de 2016 e Tombense, Uberlândia, URT e Caldense na edição de 2017 são os times em regiões mais distantes. Dessa forma, pode-se observar que o time que percorreu a maior distância está em uma região distante e o time que percorreu a menor distância está em uma região central, isso em todas as tabelas (real e gerada pela abordagem). Porém, na tabela real, especialmente os times que estão mais distantes geograficamente percorrem uma distância muito maior que os times de regiões centrais. Na Programação de Jogos gerada pela abordagem proposta, essas distâncias tendem a cair significativamente favorecendo uma competição mais justa no que diz respeito ao deslocamento.

Tabela 8. Distâncias percorridas por cada time

Time/Distância	Tabela Real		Algoritmo Proposto		Total de Distância	
	2016	2017	2016	2017	2016	2017
AME-BH/AME-BH	2408	2970	3188	3298	5552	6210
CAL/AME-TO	4338	6578	4726	5552	9286	12826
BOA/CAL	2700	6452	3130	5742	7110	11622
ATL/ATL	2708	2852	3260	3404	5552	6210
TRI/TRI	4092	5474	3540	5436	6884	9062
CRU/CRU	2844	3358	3086	3458	5552	6210
GUA/DEM	3228	6768	3308	4214	5832	10222
TOM/TOM	5414	4482	4482	5626	11094	10496
TUP/TUP	4060	5332	4782	5400	8326	9096
UBE/UBE	7628	4886	4758	5766	12354	13984
URT/URT	3892	5362	5134	5716	10086	11598
VIL/VIL	3448	2546	3352	3448	5678	6598

Adicionalmente, para demonstrar a capacidade do método de tornar mais justo o deslocamento dos times, foi utilizado o desvio padrão, que é uma medida de dispersão. Para os campeonatos de 2016 e 2017, encontraram-se os valores de 1455,47 e 1513,70, respectivamente, para as tabelas reais, e 799,71 e 1079,75, respectivamente, para a abordagem proposta. Isso demonstra que os valores distantes da média foram reduzidos pelo método proposto, como pôde ser visto pela análise do desvio padrão

A tabela real de jogos em 2017 fere uma restrição (1 - os times devem jogar em casa ou fora até duas rodadas seguidas) em contrapartida, o mesmo não acontece com as soluções encontradas pela abordagem proposta. Porém, a formulação proposta permite gerar soluções não factíveis, que atendam à organização dos jogos. Dessa forma, é possível obter maior flexibilidade de como tratar cada restrição, como permitir alguma solução não factível, exclusão ou inclusão de novas restrições.

5. Considerações Finais

Neste trabalho, foi proposta uma modelagem para o problema de programação de jogos da primeira fase do Campeonato Mineiro de Futebol. Para isso, foi utilizado o método *Simulated Annealing* juntamente com o Método do Polígono para minimizar a diferença entre a maior e a menor distâncias percorridas. Os experimentos foram realizados utilizando as instâncias dos campeonatos de 2016 e 2017. Os resultados encontrados foram comparados com as tabelas reais, geradas a partir dos jogos disponibilizados pela FMF. Em ambas as instâncias, a abordagem proposta obteve resultados melhores que as tabelas reais. Diminuindo, assim, a diferença entre a maior e a menor distâncias percorridas e criando uma programação de jogos mais justa para todos os times, principalmente para times menores que têm poucos recursos financeiros. Como trabalho futuro, pode-se destacar: 1) propor um método que permita ao método do Polígono definir também os mandos de campo, visto que o método atual gera pouca diversidade de soluções; 2) utilizar um método evolutivo populacional juntamente com o método do polígono para comparar os resultados com a abordagem proposta e com as tabelas reais.

Referências

- Anagnostopoulos, A., Michel, L., Hentenryck, P. V., and Vergados, Y. (2006). A simulated annealing approach to the traveling tournament problem. *Journal of Scheduling*, 9(2):177–193.
- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329.
- Biajoli, F. L. and Lorena, L. A. N. (2005). Uma abordagem evolutiva para o mirrored traveling tournament problem.
- Biajoli, F. L. and Lorena, L. A. N. (2006). Mirrored traveling tournament problem: An evolutionary approach. In *Advances in Artificial Intelligence-IBERAMIA-SBIA 2006*, pages 208–217. Springer.
- Biajoli, F. L., Mine, O. M., Chaves, A. A., and Souza, M. J. F. (2003). Escala de jogos de torneios esportivos: uma abordagem via simulated annealing. *XXXV Simpósio Brasileiro de Pesquisa Operacional*, pages 1295–1306.

- Biajoli, F. L., Souza, M. J. F., Chaves, A. A., Mine, O. M., Cabral, L. d. A. F., and Pontes, R. C. (2004). Scheduling the brazilian soccer championship: a simulated annealing approach.
- Capelo, R. (2017). Apesar da bonança, as dívidas do futebol brasileiro crescem – onde os lucros vão parar? <http://epoca.globo.com/esporte/epoca-esporte-clubes/noticia/2017/05/apesar-da-bonanca-dividas-do-futebol-brasileiro-crescem-onde-os-lucros-vaio-parar.html>. Acesso em: 02 junho 2017.
- Della Croce, F. and Oliveri, D. (2006). Scheduling the italian football league: An ilp-based approach. *Computers & Operations Research*, 33(7):1963–1974.
- Dinitz, J., Lamken, E., and Wallis, W. (1995). Scheduling a tournament. *Handbook of Combinatorial Designs*, pages 578–584.
- Easton, K., Nemhauser, G., and Trick, M. (2001). The traveling tournament problem description and benchmarks. In *International Conference on Principles and Practice of Constraint Programming*, pages 580–584. Springer.
- Easton, K., Nemhauser, G., and Trick, M. (2003). Solving the travelling tournament problem: A combined integer programming and constraint programming approach.
- Hoshino, R. and Kawarabayashi, K.-i. (2011). Scheduling bipartite tournaments to minimize total travel distance. *Journal of Artificial Intelligence Research*, 42:91–124.
- Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., et al. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.
- Kostuk, K. J. and Willoughby, K. A. (2012). A decision support system for scheduling the canadian football league. *Interfaces*, 42(3):286–295.
- Lim, A., Rodrigues, B., and Zhang, X. (2006). A simulated annealing and hill-climbing algorithm for the traveling tournament problem. *European Journal of Operational Research*, 174(3):1459–1478.
- Menon, V. and Jha, S. (2013). A parallel simulated annealing approach for the mirrored traveling tournament problem. *arXiv preprint arXiv:1311.1884*.
- Ribeiro, C. C. and Urrutia, S. (2007). Heuristics for the mirrored traveling tournament problem. *European Journal of Operational Research*, 179(3):775–787.
- Ribeiro, C. C. and Urrutia, S. (2012). Scheduling the brazilian soccer tournament: Solution approach and practice. *Interfaces*, 42(3):260–272.
- Segundo, C. A. B., Abreu, A. A. A. M. d., and Esmin, A. A. A. (2015). Previsão de resultados de jogos do campeonato brasileiro de futebol: Uma abordagem de mineração de dados. *XII Encontro Nacional de Inteligência Artificial e Computacional*.
- Souza, M. J. F., Mine, M. T., Silva, M. d. S. A., Ochi, L. S., and Pontes, R. C. V. (2005). Programação de jogos da primeira divisão do campeonato brasileiro de futebol por meio da metaheurística iterated local search. *Simpósio de Pesquisa Operacional E Logística da Marinha*, 8.
- Von Zuben, F. J. et al. (2002). Uma abordagem evolutiva para geração automática de turnos completos em torneios. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 13(2):105–122.

Toward a Light Version of Convolutional Neural Network for Image Classification in Embedded Systems

Cecília F. Silva¹, Clauriton A. Siebra¹

¹Informatics Center – Federal University of Paraíba (UFPB)
João Pessoa – PB – Brazil

cecilia.flavia@eng.ci.ufpb.br, clauriton@ci.ufpb.br

Abstract. *Convolutional Neural Network (CNN) is one of the most recent techniques used to image classification. However, its multiple hidden layers tend to require a high processing power, so that the most accurate implementations are not suitable to run in systems that are embedded in microcontrollers. This work aims to configure a light CNN, which can decrease its computational complexity while maintains or ever improves the classification accuracy. To that end, several pre-processing techniques and CNN architectures are defined and evaluated in a Raspberry Pi. The experiments use the German Traffic Sign Recognition Benchmark so that we could compare our results to other approaches from the literature.*

Resumo. *Redes Neurais Convolucionais (RNC) são uma das mais recentes técnicas para classificação de imagens. Contudo, suas múltiplas camadas demandam um alto poder de processamento, de modo que as implementações mais eficientes não são propícias para rodar em sistemas que estão embarcados em microcontroladores. Este trabalho tem o objetivo de configurar uma RNC mais simples, a qual tenha um menor custo computacional, mas mantenha ou até mesmo melhore a acurácia das classificações. Para isso, várias técnicas de pré-processamento e arquiteturas são definidas e avaliadas em uma Raspberry Pi. Os experimentos utilizam o banco de testes alemão de placas de trânsito, de modo que possamos comparar nossos resultados com outras abordagens da literatura.*

1. Introduction

The classification task is a traditional type of problem that makes use of supervised learning algorithms to arrange instances into different classes. Artificial Neural Networks is one of the most popular techniques for classification and its efficiency has been demonstrated in several different domains. However, some classification problems require a large amount of neurons and layers so that new strategies are being developed to improve the accuracy and efficiency of these larger structures. Deep learning is one of these strategies. It enables computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. However, these structures generally increase the need for computational power and memory, so that they are not able to efficiently run in limited platforms, such as microcontrollers.

The focus of our work is on image classification and it is part of a larger project that intends to provide levels of autonomy to vehicles. Our current task is to create a

module that is able to classify traffic signs. For example, after a traffic sign is recognized, our module must classify it as a sign of maximum velocity or any other. The literature shows that the best approaches for this kind of problem are based on Convolutional Neural Networks (CNN), which is one example of deep learning technique. In fact, CNN was used by the winner team [Masci et al. 2012] of the German Traffic Sign Recognition Benchmark competition [Salmen et al. 2011], which obtained an accuracy of 99.46%.

The aim of our work is to propose a CNN method that presents a similar accuracy to the best current approaches, while reduce its computational costs (processing power and memory), since we intend to run all modules of our project in a microcontroller Raspberry Pi. Another very important constraint is the processing time. The identification of the traffic sign must be carried out in real time and such identification must be processed in less than 1 second after the sign recognition. This time constraint may also be a problem to other approaches, since better results generally requires higher processing times.

The remainder of this paper is organized as follows: Section 2 presents the related works, stressing the best current approaches for image classification and the limitations of these approaches if we consider their use in limited platforms. Section 3 shows our research method and the sequence of strategies that were used to improve the results of our CNN. Section 4 discusses the results of our experiments using the German Traffic Sign Recognition Benchmark (GTSRB). Finally, Section 5 concludes this work with the main remarks and research directions.

2. Related Work

Previous works have also focused on the problem of traffic signs classification using popular CNN architectures. The work of [Nardi et al. 2016] proposed a multi-scale CNN based on LeNet CNN [LeCun and Sermanet] and VGGNet [Simonyan and Zisserman 2015] works for classification of images, where VGGNet is a popular choice for feature extraction of images. This CNN has a very uniform architecture with 16 convolutional layers, all of them with 3x3 convolutions and 2x2 pooling. However, although it is easy to implement due to its uniformity, this CNN has 140 million parameters, resulting in one of the most computational expensive architectures. Differently, LeNet has a simple architecture with three convolutional layers. The resultant CNN was evaluated in both GTSRB and Italian Traffic Signs (DITS) datasets, achieving an accuracy of 98.2% in the former dataset.

The work of [Ucar and Demir 2016] proposed a comparison between approaches using the AlexNet CNN, SURF [Tuytelaars et al. 2004] and HOG [Dalal and Triggs 2005]. This CNN had a total accuracy of 92.80%, while SURF and HOG approaches had accuracies of about 82.90%.

The work discussed in [Masci et al. 2012] used a 9 layer CNN in the same fashion than the one proposed in the LeNet CNN. This work was the winner of the German Traffic Sign Recognition Benchmark competition [Salmen et al. 2011] with accuracy rate of 99.46%. On this competition, [LeCun and Sermanet] also applied different configurations of their CNN network and achieved 98.31% of accuracy. However, different improvements were carried out in their work after the competition by using grayscale images, rather than all RGB channels, and by reducing the hidden layers of the network. After that, an accuracy of 99.17% was obtained. Other works based on HOG and Cosfire

color-based filters [Petkov 2007] respectively obtained 95.68% and 98.97% of accuracy in the GTSRB domain.

The approach with the best GTSRB accuracy was described in [Rowen et al. 2016], which developed a hierarchical approach with six CNNs. In this approach, the 43 GTSRB classes were divided into five groups and the first CNN was responsible to decide which group the received traffic sign belongs. Thereafter, one of the remaining five CNNs, corresponding to the detected groups, carries out the traffic sign classification.

Table 1 summarizes all the models and their accuracies, which were based on experiments using the GTSRB dataset. This table also shows the human performance regarding this dataset, which was presented in [Salmen et al. 2011] for comparison.

Table 1. State of the Art Methods and their Accuracies

Author	Method	Accuracy
Rowen et al, 2016	Hierarchical CNNs	99,58%
Masci et al, 2012	CNNs	99,46%
LeCun and Sermanet, 2011	Reduced CNNs	99,17%
Petkov et al, 2017	Color-blod-based COSFIRE filters	98,97%
Salmen et al, 2012	Human performance	98,84%
LeCun and Sermanet, 2011	CNNs	98,31%
Nardi et al, 2016	Multi-Scale CNN	98,20%
Hamdoun et al, 2011	Random Forests	96,14%
Salmen et al, 2012	LDA on HOG 2	95,68%
Salmen et al, 2012	LDA on HOG 1	93,18%
Salmen et al, 2012	LDA on HOG 3	92,34%

Any of these approaches were evaluated in microcontrollers' platforms. Rather, powerful hardware architectures, such as the NVIDIA Tesla GPU Accelerators for Servers, are generally used to ensure short times. The first place in this table [Rowen et al. 2016] is in fact a proprietary and non-public approach. This solution runs in a very specific hardware (Tensilica Vision P5 DSP - Digital Signal Processor), which is a high-performance DSP specifically designed for image and computer vision processing. Authors have developed another algorithm for the performance-versus-complexity tradeoff and they have been able to achieve a reduction by a factor of 86%, but with a correct detection rate degradation of about 2%. This degradation means a final result of 97.58%, which is lower than the human performance. The paper regarding the second place [Masci et al. 2012] uses a system with a Core i7-950 (3.33GHz), 24 GB DDR3, and four graphics cards of type GTX 580. Unfortunately this work does not indicate time results, since they are only worried about the performance of their approach. Thus, based on such works, we see that the current approaches are not proper to run in limited platforms, such as a Raspberry Pi microcontroller.

3. Research Method

3.1. Benchmark Setup

The German Traffic Sign Recognition Benchmark (GTSRB) dataset, proposed on the 2011 IEEE International Joint Conference on Neural Networks [Salmen et al. 2011] and widely used in many conferences and papers, was chosen to be the benchmark of our experiments. GTSRB has a strong variation in visual appearance of signs regarding illumination, weather conditions and partial occlusions (Figure 1a). The dataset consists of 39209 (32x32px) color images for training and 12630 images for testing; which are divided into 43 different imbalanced classes (Figure 1b).

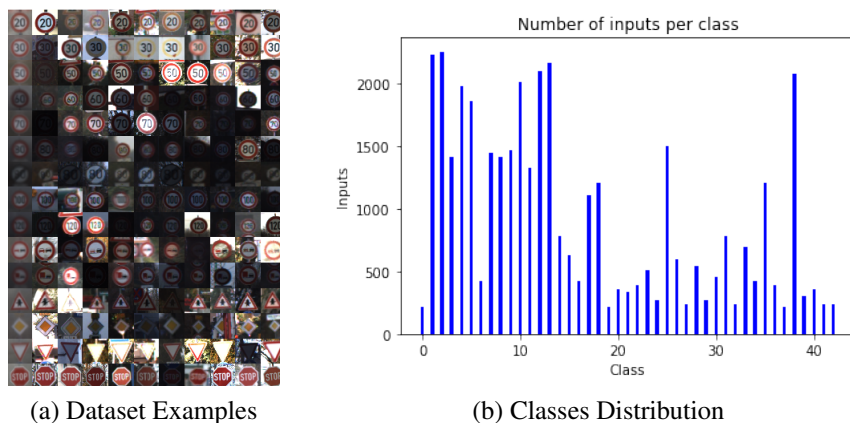


Figure 1. Examples of instances from the GTSRB and distribution of such instances in 43 classes, each of them representing a traffic sign.

To better balance the number of instances in each class, we have oversampled some classes. This process was carried out in two phases. First, some images were rotated using 8 different angles (-5, 5, -10, 10, 15, -15, -20, 20) along their own axis. These angles represent approximations of possible problems in the position of signs. Then, standard deviation was used as normalization technique for images brightness, so that they were scaled to the interval of [0,1]. The resultant dataset was used along the next stages of our research.

3.2. Reference Model Architecture – CNN₁

The next stage of our method was to define a very simple architecture (CNN₁ - Figure 2), so that we could have a reference for more complex architecture regarding different evaluation parameters.

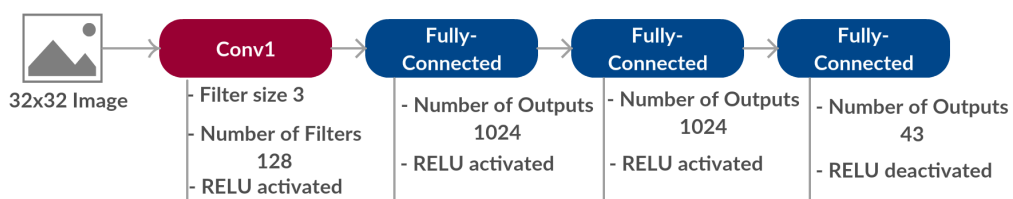


Figure 2. Structure of CNN₁

Then, a simple model was defined with only one convolutional layer and three fully-connected (FC) layers. Regarding the parameters in each layer, the first layer (Conv1) receives an image as input and has a depth of 128 filters to process three times each image, considering that each image has a 32x32 dimension. Each filter has a size equal to 3x3 to simplify the model. In this stage, all the color channels (RGB) were considered and hyper-parameters zero-padding and stride had their values set respectively to true and 1x1. The output of this layer feeds the first FC layer, resulting in 32x32x128 inputs (the size of Conv1 depth multiplied to image dimension) and 1024 outputs, which is the total number of pixels in each image. The second FC layer has 1024 inputs and 1024 outputs and both layers use RELU activation function. The final FC layer is only for classification, so that it has 1024 inputs and 43 outputs (the amount of classification classes). This stage does not have an activation function.

3.3. CNN Improvements – CNN₂

The next stage was to implement a more complex CNN (CNN₂ - Figure 3). Rather than using only one conv layer with 128 filters, three layers were considered with 32, 64 and 128 filters respectively. The main idea behind this modification was to get the more detailed features of the image by first using the total number of features equal to the image width. After that, the CNN captures more detailed features by using the double and triple number of filters. In addition, all layers have each filter of size 5x5 instead of 3x3 to get more feature information and to increase accuracy rates. RELU was again used as activation function. However, a high number of convolutional layers results in a large quantity of parameters and outputs. Thus, pooling layers with stride equal to 2x2 were added after each conv layer to downsample this quantity and to reduce overfitting. This strategy was based on the LeNet architecture [LeCun and Sermanet]. Moreover, each pooling performs a dropout regularization [Srivastava et al. 2014] with a probability of 0.5.

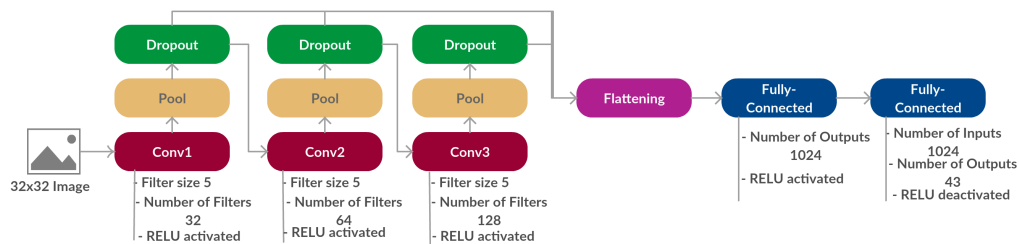


Figure 3. Structure of CNN₂

The next layers, Conv2 and Conv3, have their inputs equal to the number of features after the dropout action of the previous layer. Then, the three layers with dropout are concatenated in a flattened layer to feed the FC layers. In this final stage, the first FC layer has a total of 1024 outputs and RELU activation function is used. The last layer is just for classification and it has the same configurations than the last layer of our first proposed architecture (CNN₁).

The initial experiments with CNN₂ used the same dataset than CNN₁. This means, no special preprocessing technique was applied. The idea was to isolate the improvements cause by the new configuration of CNN₂ regarding CNN₁.

3.4. Dataset Improvements –Balance and Preprocessing

The dataset is so important as the own model. Thus, we focused this stage of our research on its improvement. This first action was to better balance our dataset. To do that, affine transformations were applied to the original dataset, according to Algorithm 1, so that new images could be generated (oversampling). These transformations were implemented to simulate different effects on the signs viewing on different angles and distances. Hence, translation is applied to consider viewpoints, so the signs do not need to be very close to be recognized. Rotation is also considered to simulate situations where the traffic sign or detection camera presents variations due to environment changes or smooth curves in the road. In this algorithm, each instance produces another new instance with rotation, translation and shear variables randomly assigned to a value between $[-30,30]$, $[-5,5]$ and $[0,5]$, respectively; until the maximum number of extra features allowed for each class is obtained.

Algorithm 1 Affine Transformations to Oversample Data

```

1: for  $i \leftarrow 0$  to  $totalNumberClasses$  do
2:   for  $feature \leftarrow 0$  to  $featuresInTheClass[i]$  do
3:      $FeatureLimit = 0$ 
4:     if  $inputsPerClass[i] \geq 1800$   $inputsPerClass[i] \leq 2500$  then
5:       if  $FeatureLimit \leq inputsPerClass[i]/3$  then
6:          $generateNewFeature(feature, rotation, translation, shear)$ 
7:          $FeatureLimit+ = 1$ 
8:       end if
9:     else if  $inputsPerClass[i] \leq 1300$  then
10:       $generateNewFeature(rotation, translation, shear)$ 
11:      if  $FeatureLimit \leq inputsPerClass[i]/2$  then
12:         $generateNewFeature(feature, rotation, translation, shear)$ 
13:         $FeatureLimit+ = 1$ 
14:      end if
15:    else
16:       $generateNewFeature(feature, rotation, translation, shear)$ 
17:    break
18:  end if
19: end for
20: end for
21:

```

After the benchmark balance, several preprocessing techniques were applied on each of its instances. In order, studies that investigate image classification agree on the almost compulsory use of preprocessing techniques to adequate some images features to training algorithms. This stage of our method applied some of these techniques.

The first technique was a histogram equalization, which was applied to all RGB channels to let all pixels more uniform, considering that some images are darker than others. As the application of different segmentation techniques can considerably produce different results [Masci et al. 2012], we also evaluated the histogram equalization performance by comparing it to the adaptive histogram equalization [Brown and Ballard, 1982].

Second, images were converted to grayscale so that their pixels could be represented by 8 bits instead of their original 32 bits. This process mainly reduced the data average volume. Furthermore, other works obtained better results when they used grayscale images rather than RGB color-channels [LeCun and Sermanet]. The image was also zero-centered using mean subtraction. Third, the first stage of our method (Section 3.1) used standard deviation as normalization technique, so that all images were scaled to an interval of [0,1]. Differently, in this stage we used min-max normalization to scale the pixels to an interval of [-5,5], so that the CNN could better differentiate the pixels, but they still limited inside a shorter scale rather than from 0 to 255. Finally, the dataset was shuffled to avoid that the network learns with successive highly correlated features, thus reducing generalization errors.

After the application of these oversampling and preprocessing methods, the training data was randomly divided into training and validation data so that the 5-fold cross validation could be applied.

3.5. Model Refining – CNN₃

A final modification was carried out in CNN₂ with the insertion of an extra pool layer in the convolution layers 1 and 2 to downsample the features and minimize the errors (CNN₃ - Figure 5). As the first layer is the one with the higher level of noise and less refinement data, the pooling was defined with a 4x4 dimension, while the second layer had a 2x2 dimension. Since the convolution layer 3 had the most important features, no additional pooling was performed.

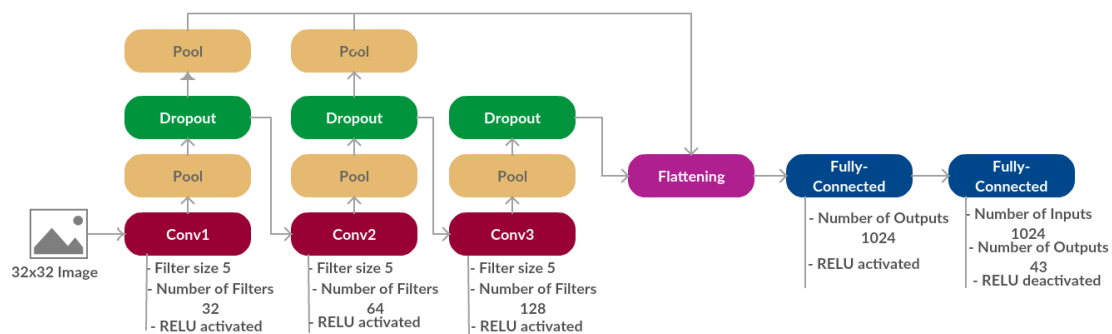


Figure 4. Structure of CNN₃.

Moreover, this phase used early stopping [Yao et al. 2007] as regularization technique, with a patience of 10000 iterations. If the best accuracy rate is not changed thereafter, the model is saved for further use. The parameters and respective values for this last CNN training were: L2 regularization with epsilon = 0.0001; dropout with keep probability = 0.5; early stopping with patience = 10000; AdamOptimizer[Diederik and B. 2015]; total number of epochs = 100; and learning rate = 0.0001.

3.6. Evaluation Method

The three previous CNNs were implemented in Python using the Native TensorFlow (TF) framework [Barham et al. 2015]. After that, CNN₃ was chosen to be implemented in two more framework configurations: TensorFlow with Keras and Theano with Keras. Theano

is an alternative to TF and it is indicated as one of the Deep Learning frameworks with the best results [Bahrampour et al. 2016]. The Keras [Chollet 2015] API implements both TF and Theano frameworks using optimization techniques and providing a higher level of abstraction to final uses. The aim in using these three configurations (Native TF, TF with Keras and Theano with Keras) was to verify if such configurations have impact on the parameters in evaluation.

After running the training experiments in such frameworks, results regarding accuracy were compared to verify the best configuration. These experiments first used a high-end PC with i7 processor of 3.0GHz and 8GB of RAM memory. Then, the best configuration (CNN₃) was uploaded in a raspberry PI 2B with 1Gb RAM and a 900MHz quad-core ARM Cortex-A7 processor. The idea of this stage was to measure the hardware performance using the next metrics:

- Time to load a model: after the training stage, Native TF has the option of saving its model with not only its weights, but all the variables and the full network graph. This process results in a very large model. Keras, on the other hand, saves both models (TF or Theano) in a simpler file with only its network weights;
- Total prediction time: time to classify a single image;
- Memory and CPU usage: previous works do not give so much emphasis to such parameters, since powerful hardware are used along their experiments. However, they become very important in limited platforms, mainly when we consider that other resources (e.g., camera, driver control) will also use such platform, which needs to share their resources;
- Temperature: this metric is important because high variations of temperature have negative impacts on embedded systems.

Results regarding this evaluation process are presented in the next section, together with on the fly modifications that were carried out when we identified opportunities to improve the final results.

4. Results

4.1. CNN₁ - One Convolutional Layer

The first CNN model (CNN₁), which was used as reference, obtained an accuracy of 99.8% in the training validation dataset using the 5-fold cross-validation method. However, the use of this model in the test dataset only resulted in an accuracy of 91.7%. This value is lower than any other model in Table 1, which demonstrates the weakness of the CNN₁ model. Our analysis of this model showed that the overall number of classifications for each instance was correct. However, “speed limit” signs were the classes with the highest rate of error. Other classes with considerable outliers were the signs with triangular format (e.g. general caution, dangerous curve, pedestrian and bicycles crossing). In general, these classes have similar shape and symbols, which cause problems regarding the creation of models.

4.2. CNN₂ - Three Convolutional Layers

The analysis of the previous results showed that CNN₁ was not correctly processing images of specific geometrical shape and with digits (e.g. speed limit signs). Then, the

number of convolutional layers was increased to three, so that the network could refine the features acquired from the images. However, as the previous model was already presenting overfitting issues and the increase regarding the number of convolution layers could enhance such overfitting, dropout and L2 regularization techniques were applied in addition to pool layers after each convolution.

Considering these changes, the model accuracy increased to 94.4% (test dataset), while such accuracy was about the same in the training and evaluation dataset (99.4%). The analysis of the results showed that the changes in the model were efficient to reduce the problems related to geometric shape and just some isolated cases remained. However, classification regarding speed limit classes was still presenting a high rate of error.

4.3. CNN₂ with Data Balancing and Preprocessing

The CNN₂ model presented a good improvement regarding CNN₁, so that this same model was used again. However its training dataset was modified according to Section 3.4. The better balance of the number of instances in each class (minimum value of 2500 and maximum value of 3000), as illustrated in Figure 5, together with the preprocessing techniques, increased the accuracy to 96.7% (test dataset).

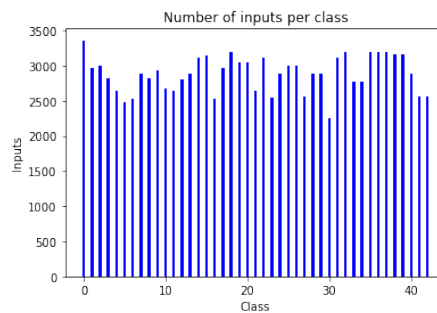


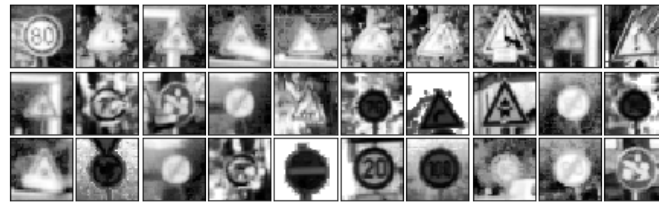
Figure 5. Distribution of instances at the final of the oversampling process.

The analysis of this new result showed that this change in the dataset had a significant positive impact on the speed limit signs. However, two specific classes, which were not previously presenting classification problems, started to have their instances erroneously classified in each other. Then a new analysis was carried out in such results, but, at this time, in the images of the datasets after the histogram equalization. This analysis showed that images with false classifications had a darker background with very light traffic signs, causing a loss of important features regarding the symbols at the center of such signs. Figure 6a shows examples of signs that were erroneously classified.

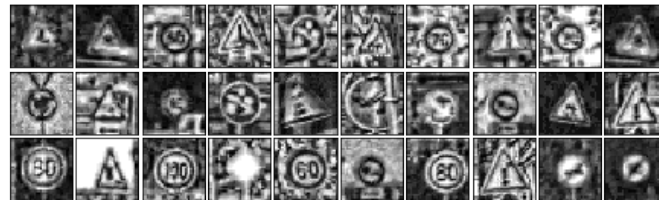
Based on this analysis, we identified the opportunity to test a different form of equalization (adaptive equalization). Using such strategy, the validation accuracy increased to 97.7% (test dataset). In fact, examples of images erroneously classified, after the use of the adaptive equalization, showed that such images were not still presenting such problem (Figure 6b).

4.4. CNN₃ – Improved CNN₂

The accuracy of our previous model (97.7%) was still lower than the average accuracy of human recognition (98.84%), according Table 1. Thus, we tried two further optimization



(a) Histogram Equalization



(b) Adaptive Histogram Equalization

Figure 6. Appearance of some instances after the use of different equalization methods.

strategies on the network to improve our accuracy ratings, as discussed in Section 3.5. The use of an extra pool layer, to reduce the noise, increased the accuracy to 98,1%. As the model could overfit after some epochs, early stopping regularization was our second and final change. Then, the network obtained a maximum accuracy of 98.22% after 23 epochs and it did not improve after 10000 interactions. Thus, the final model used only those 23 epochs for training instead of 100. In addition, learning rate was set to 0.00003 instead of 0.0001, since the learning curve was decaying too fast over the first two epochs.

Our analysis also showed that this model did not present issues related to a specific type of traffic sign and the error was more distributed over all instances. This means, this pattern or error was not identified in our analysis regarding the instances that were erroneously classified. It is important to notice that we could use more complex networks, as used by the best approaches in Table 1. However, this was not our intention since we intend to have a model as lighter as possible. Thus, we need to consider the tradeoff accuracy and hardware performance. Next section discusses our experiments regarding this latter aspect.

4.5. Analysis of Hardware Performance

The main aim to evaluate our model with 98.22% of accuracy on Native TensorFlow, TensorFlow with Keras and Theano with Keras; was to verify which configuration is the most feasible choice considering accuracy ratings and hardware overall performance. In this way, Table 2 shows that Keras outperformed Native TensorFlow due to its code optimizations, obtaining an accuracy of 98.57% in both frameworks, while Native TensorFlow maintained the same original accuracy of 98.22%.

Regarding the hardware performance (Table 3), Native TensorFlow spent about 51 seconds to load the model into a Raspberry PI 2B. TensorFlow with Keras spent 6.36 seconds to load the same model, while Theano with Keras spent 13.61 seconds. This loading time for Native TF could be explainable considering that this framework restores all network structure and its variables, resulting in a 40Mb model. Differently, Keras produces a model only with its weights, resulting in a file with 9.1Mb for both frameworks.

Table 2. Hardware Performance and Accuracy Analysis

Model	Accuracy(%)	Memory Usage (mb)	Model Size (mb)
Native TensorFlow	98.22%	296mb	40mb
TensorFlow on Keras	98.57%	408mb	9.1mb
Theano on Keras	98.57%	401mb	9.1mb

Table 3. Hardware Performance Analysis

Model	Load Model Time(s)	CPU Usage (%)	Classification Time(s)
Native TensorFlow	59.58	26%	0.9
TensorFlow on Keras	6.36	30%	0.72
Theano on Keras	13.61	29%	6.49

Although Native TF spent 51 seconds to load its model, its classification was carried out in only 0.95 seconds. This is a relevant result to the autonomous vehicle domain, since several other functions will only be carried out after this classification (e.g. tracking and driver decision). The classification in Theano spent 6.49 seconds and only 0.72 seconds on TensorFlow with Keras. This was the best classification time of all three models. Table 2 and 3 illustrate the overall rates using these metrics in addition to CPU usage, which is around 25-30% in Raspberry PI 2B CPU. This is the overall usage for most actions on this embedded system. Temperature also did not present a considerable variation and it is maintained in around 35°C.

Another important conclusion is about the memory usage. As Native TensorFlow saves its full model graph and variables in memory, its model size reaches 40 Mb, while the two other approaches only use 9.1Mb. On the other hand, its RAM memory usage is the lowest (296 Mb). Considering that the Raspberry PI 2B has 948 Mb of total RAM memory and its usage for common processes is around 195 Mb, any model expansion, such as the use of a higher number of layers, could create problems regarding the memory use. Thus, all these tradeoffs must be considered in the final configuration.

5. Conclusions and Research Directions

The final proposed CNN architecture has three convolutional layers and its best result (98.57% of accuracy) was obtained using Keras API with both TensorFlow and Theano implementations. However, Theano with Keras has a prohibitive classification time, so that this approach is not proper to our domain. This result is still slightly lower than the human recognition accuracy (Table 1), however its classification time is lower than 1 second running in an embedded system and this is a fundamental requirement of our domain. Our future challenge is to improve the accuracy of our model without increasing its hardware requirements, since the use of memory is already higher than 400 Mb. This is almost the half of the available memory of the Raspberry PI 2B.

Another important contribution of this work is the discussion about the whole process used to obtain the current model. We indicated the modifications in the model, their rationale and results obtained with such modifications. Thus, the lessons learned from these practical experiments could also support other researchers that intend to create similar models.

References

- Bahrampour, S., Ramakrishnan, N., Schott, L., and Shah, M. (2016). Comparative study of caffe, neon, theano and torch for deep learning. International Conference on Learning Representations.
- Barham, P., Abadi, M., and Agarwal, A. (2015). Tensorflow: Large-scale machine learning on heterogeneous distributed systems.
- Chollet, F. e. a. (2015). Keras. In <https://github.com/fchollet/keras>.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, pp. 886-893.
- Diederik, P. K. and B., J. (2015). Adam: A method for stochastic optimization. 3rd International Conference for Learning Representations, San Diego.
- LeCun, Y. and Sermanet, P. Traffic sign recognition with multi-scale convolutional networks. Neural Networks (IJCNN), The 2011 International Joint Conference on, 2809-2813.
- Masci, J., Schmidhuber, J., Cirefisan, D., and Meier, U. (2012). Multi-column deep neural network for traffic sign classification. Neural Networks Journal, Elsevier Ltd.
- Nardi, D., Bloisi, D., Youssef, A., and Albani, D. (2016). Fast traffic sign recognition using color segmentation and deep convolutional networks. Advanced Concepts for Intelligent Vision Systems. Lecture Notes in Computer Science, vol. 10016, pp. 205-216.
- Petkov, N., G. B. A. G. (2007). Color-blob-based cosfire filters for object recognition. Image and Vision Computing, vol. 57, pp. 165-174.
- Rowen, C., Hijazi, S., and Kumar, R. (2016). Using convolutional neural networks for image recognition. Cadence.
- Salmen, J., J., S., M., S., and Igel, C. (2011). The german traffic sign recognition benchmark: A multi-class classification competition. International Joint Conference on Neural Networks.
- Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, pp. 1-14.
- Srivastava, N., Hinton, G., Krizhevsky, A., and Sutskever, I. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research.
- Tuytelaars, T., H., B., and van Gool, L, I. (2004). Surf: Speeded up robust features. International Journal of Computer Vision.
- Ucar, A. and Demir, Y. (2016). Moving towards in object recognition with deep learning for autonomous driving applications. International Symposium on Innovations in Intelligent Systems and Applications.
- Yao, Y., Rosasco, L., and Caponnetto, A. (2007). On early stopping in gradient descent learning. Constructive Approximation. 26 (2): 289-315.

Save the Data! An Intelligent Approach to Avoid Data Loss

Marcos Iseki, Bruno Nogueira, Brivaldo Junior¹

¹Faculty of Computing (Facom) – Federal University of Mato Grosso do Sul (UFMS)
79.070-900 – Campo Grande – MS – Brazil

{iseki, bruno, brivaldo}@facom.ufms.br

Abstract. *Data loss can harm customers, business strategies and companies reputation. While enterprise environments commonly employ data replication technologies as RAID, small business and customers rely on the lifetime of their storage devices, mostly hard drives. Thus, as these hard drives fail, massive data losses may happen. When important data is at stake, being aware of possible disk fails is crucial. In this sense, hard disks use SMART technology to try to detect failures. These analysis, however, are carried out only when operational system requires or during boot process. Moreover, these predictions are not very accurate, presenting small accuracies and high false positive rates. To avoid such problems, we propose a machine learning approach to detect hard drive failures. We use a huge and recent dataset from Blackblaze. Decision trees achieved the best performance with 80% in accuracy rate and less than 12% in false positive rate in failure predictions.*

1. Introduction

Failures in computer hard disk drives (HDDs) may cause significant data loss. Either permanent or temporary, data losses are normally associated with costs of availability and working time, not to mention the importance of data itself. In 2010, industrial level sectors like energy, telecommunications, or manufacturing could have a lost revenue that ranged between \$1.6 and \$2.8 million per hour of computer downtime, or data loss [Kroll Ontrack Inc. 2010]. For small businesses, that could mean their business survival, and for customers, the loss of important personal data.

Despite backup being a typical countermeasure against that, in 2016, around 24% of American computer users never replicated all data [Klein 2016]. That percentage reaches 66% if we add yearly and less often distributions. On the opposite side, only 8% of users backed up personal information once a day or more often, meaning not only that a large amount of data is in danger of loss, but that the associated risks and costs are not widespread yet.

At the same time, cloud storage fees are still high, mainly for small businesses and customers. Considering average prices of services in the first quarter of 2017, in less than two years, a client paying for a 1 TB storage service could purchase a brand new external HDD of same capacity.

In the high scale scenario, data center storage installed capacity doubles every year since 2015. Cisco Systems predicts that, by 2020, the amount of data stored in data centers and on client devices will reach 915 EB and 5.3 ZB, respectively [Cisco Inc. 2016]. Also, every year, HDDs' vendors sell hundreds of millions of devices, being HDDs the most common computer storage appliance for both enterprise and customer segments.

Another survey conducted by Kroll Ontrack Inc. revealed some causes of data loss and the impact to businesses and home users [Kroll Ontrack Inc. 2014]. The survey was carried out in ten countries across North America, Europe and Asia Pacific, in 2014. Hardware crash or failure was accountable for 66% of data loss, and within that segment, HDD crashes for 72% of imputed causes. Individual response segments also confirmed HDD failure as the main reason of data loss, among 71% of businesses and 72% of home users.

Hard disk Annualized Failure Rate (AFR) is a disk failure probability within one year of full use. It is calculated by manufacturers, and specified in device datasheets, being an important parameter to calculate drive reliability. Manufacturers get this metric during stress tests, reporting a scaled score. Data centers and other companies that work with large HDD population may calculate their annual failure rate, calling it either AFR or Annual Replacement Rate (ARR). Hence, ARR is not an estimation, but the yearly ratio between failed drives and total number of a drive population.

Schroeder and Gibson confirmed a disagreement between datasheet AFRs, and ARRAs acquired from field-gathered disk replacement data [Schroeder and Gibson 2007]. The authors analyzed a set of 100,000 HDDs from different vendors, running different systems, and interfaces (SCSI, FC, and SATA). Nominal AFRs ranged between 0.58% and 0.88%. However, ARRAs started from 0.5%, reaching 13% in some systems, more than 14.7 times higher than stated in datasheets.

Besides, failure rates are expected to follow a pattern known as “bathtub curve”, which means both high early-failure (also referred to as infant mortality rate) and wear-out rates, and in-between a low constant failure rate, as shown in Figure 1. However, Schroeder and Gibson also confirmed that ARRAs increase within a period of five years, contradicting the low constant failure rate. Pinheiro [Pinheiro et al. 2007] also verified similar analysis over a different dataset, as we will discuss on Section 2.

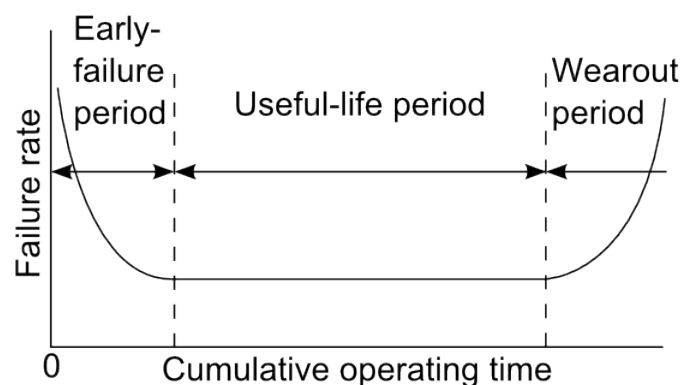


Figure 1. HDD failure pattern, or the “bathtub curve” [Yang and Sun 1999].

After a HDD crash, there are some cases when data retrieval is possible through drive recovery, but cost of services are considerably high. It changes mainly according to severity of the damage, with an average price of \$1,325 [Dell inc. 2016] per drive, at least 4-5 times more expensive than a brand new HDD.

When even backups may fail, **Self-Monitoring, Analysis, and Reporting**

Technology (SMART) is a feature that allows failure prediction of SMART-capable drives. SMART is an Advanced Technology Attachment (ATA) standard since 2003. In the current version, the system works by taking a snapshot of 45 device health parameters, comparing them with vendor specific thresholds. At operational system boot screen, SMART is able to alert the user when a failure is imminent, providing an estimated 3-10% detection rate with 0.1% false alarms [Murray et al. 2005].

This detection rate could be significantly improved using machine learning algorithms [Mitchell 1997]. Previous works exploit the construction of predictive models in order to anticipate disk failures using SMART information presenting significantly better results in terms of precision and false alarms [Murray et al. 2005, Pinheiro et al. 2007, Pitakrat et al. 2013]. Some of these works, however, are focused on predicting failures in specific HDD models. Besides, disk failures evolve as the technology evolves. Thus, some predictive models may be now outdated.

From these considerations, in this work we evaluate whether different machine learning algorithms increase accuracy classification of impending failures of diverse SMART-capable devices. We compare the models by their accuracy and execution time. We also intend to test whether our findings either validate or contradict previous work found in the literature.

This paper is organized as follows. In Section 2 we discuss some previous related works. In Section 3 we describe the dataset. In Section 4 we describe the HDD failure prediction model, and analyze the gathered results. Finally, in Section 5 we highlight conclusions, and future works.

2. Related work

Murray et al. analyzed a set of 68,411 instances of 64 HDD attributes from a specific vendor, including the class value, which can be either *good* or *failed*, represented as boolean values [Murray et al. 2005]. Each instance has SMART logs collected every two hours from 369 single model HDDs, with 191 failed units, and 178 good ones. The authors compared the performance of two machine learning models: SVM and Multiple Instance Naïve Bayes (mi-NB). The Support Vector Machines achieved 50.6% in detection rate (accuracy) and 0% in false alarm. On the other hand, mi-NB achieved 34.5% in detection rate and 1.0% in false alarms. They mentioned that SVM was computationally expensive, although it had an excellent false positive rate. Also, mi-NB was suitable for on-line tests due to lower memory and computation requirements.

We were able to identify at least three weaknesses in Murray's study, two regarded to dataset, and one to result. It is important to mention that these weak points are opportunities to expand research in this area, and are not necessarily mistake notations. First, the dataset covered a small collection of HDDs all from the *same model*, possibly reducing capacity of the trained algorithms to interpret test samples from different models. Second, SMART logs of failed drives were taken *after* drives were declared failed by customers. This means that the algorithms learned to classify drives after they failed. The dataset we used covers 41,980,868 entries, 55 different HDD models from five major manufacturers. All data related to failed drives were taken on their last day in operation.

Finally, when we analyze results, it is possible to say that 50.6% or 34.5% detection rate for binary classification is statistically close to randomly categorize the object in

focus, like flipping a coin. In this study, we have reached both higher accuracy and less computational time.

Pinheiro et al. studied failure patterns of HDDs in a large disk drive population from different manufacturers, all running some of Google's systems [Pinheiro et al. 2007]. Google developed a System Health infrastructure which basically collected, stored, and mined data from servers and other data bases. The time-series information includes environment temperature, utilization rate, repair events, and some SMART parameters.

The data analysis revealed four important findings: first, either temperature or activity level have little correlation with failure rates; second, a total of four SMART parameters had a large impact on failure probability, so that drives are significantly more likely to fail after an occurrence of those SMART variables; third, AFRs does not follow the "bathtub curve", as manufacturers expected; and fourth, 36% of all failed drives had zero counts on all SMART variables. Based on this third finding, the authors concluded that SMART data alone is not effective to build HDD failure prediction models. The results we have found, though, add a new comprehension over SMART parameters, and their contribution to HDD failure prediction.

Pitakrat et al. preprocessed the dataset used in Murray et al. [Murray et al. 2005], classifying instances into two classes, 7 days before failing (as DAY_7 class) and not failing (as INF class) [Pitakrat et al. 2013]. They also selected 26 out of 64 original parameters, eliminating drive serial numbers, hours of operation, and other constant attributes. The authors compared prediction quality and time required to build training and prediction phases of 21 machine learning algorithms. The results suggested that algorithms with high prediction quality are nearest neighbor classifier, random forest, C4.5, REPTree, RIPPER, PART, and K-Star (accuracy above 97%). Those suitable for online predictions are Bayesian network and OneR (less than 0.2s demanded in prediction phase and 89% accuracy). And those with very low false alarm rates are SMO, and SVM.

In this case, we were able to use the dataset to reproduce some aspects of the original training-prediction experiment, achieving good results in terms of accuracy and processing time, as shown in Table 1. However, as all HDDs were from the same manufacturer, SMART parameters were identified by specific names that did not match with SMART logs we can acquire from different HDDs now. Hence, we could not verify whether the algorithms would disclose similar results when facing updated datasets.

Table 1. Performances of Machine Learning Algorithms considering 68,411 inputs from Murray's dataset. Results represent the average of ten runs, with ten folds.

Machine Learning Algorithm	Accuracy	Time (sec)
Naive Bayes (NB)	0.86±0.03	0.08
Multinomial Naive Bayes (MNB)	0.84±0.06	0.12
Multilayer Perceptron (MLP)	0.87±0.00	0.51
Decision Tree (DT)	0.80±0.19	0.73
KNN Euclidean (KNN)	0.77±0.16	0.91
Logistic Regression (LR)	0.84±0.08	1.75
SVM with kernels (SVM)	0.87±0.00	772.71

Another obstacle we found was evaluating the effectiveness of data preprocessing. The authors stated that some instances were reclassified as DAY_7, meaning that the algorithm would be able to detect or anticipate a HDD impairment 7 days before it actually happens, allowing the user to plan the device replacement with some success. Nevertheless, the main problem is that SMART stats that could identify failures were taken from HDDs *after* they were declared failed, as we have mentioned before.

Given this scenario, in this work we carried out a comparison of some of the state-of-the-art classifiers in predicting disk failures. These algorithms were applied in a recent dataset containing data of multiple HDD vendors which is described in the next section.

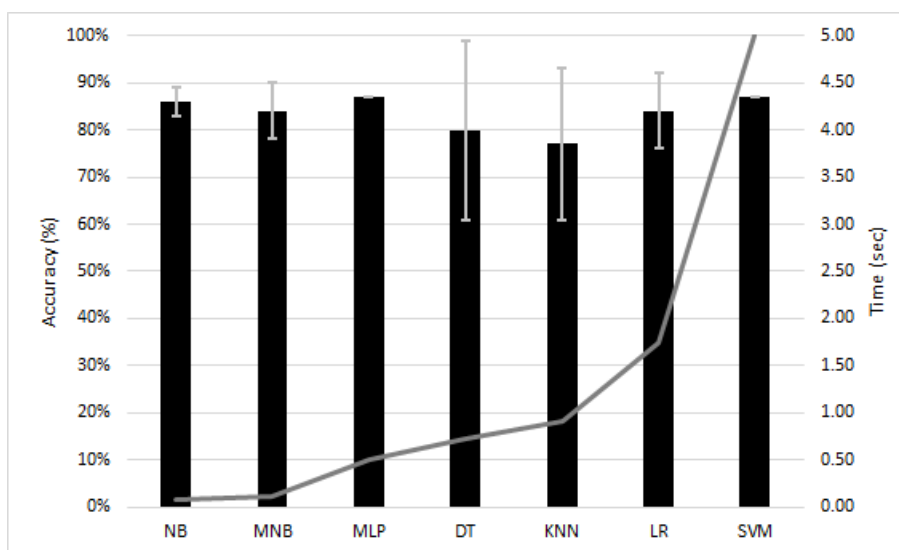


Figure 2. Representation of Table 1. The execution time of SVM was cutoff.

3. Dataset Description

Backblaze Inc. updates the dataset used in this study since April, 10 2013, available with free access on their website¹. It consists of daily files in comma-separated values (CSV) format.

Each file has a set of SMART logs of every HDD in operation in that specific day. The number of HDDs in use shifts occasionally as the company discards defective devices, adding and/or replacing drives. By the end of 2013, Backblaze had 27, 223 HDDs in operation, increasing this number to 73, 653 in 2016, a 271% change. During the same period of four years, they kept an average AFR of 3%, as shown in Figure 3, which is 2.2 to 6 times higher than datasheet AFR.

Files ranging from 2013 to 2014 have 80 columns related to 40 different SMART stats, and those ranging from 2015 have 90 columns related to 45 different SMART stats, with both normalized and raw values. Because of that difference in the number of SMART stats, and due to general SMART data quality, we have decided to use only datasets of 2015 and 2016 in the training and prediction of algorithms.

¹The dataset can be downloaded on <https://www.backblaze.com/b2/hard-drive-test-data.html>

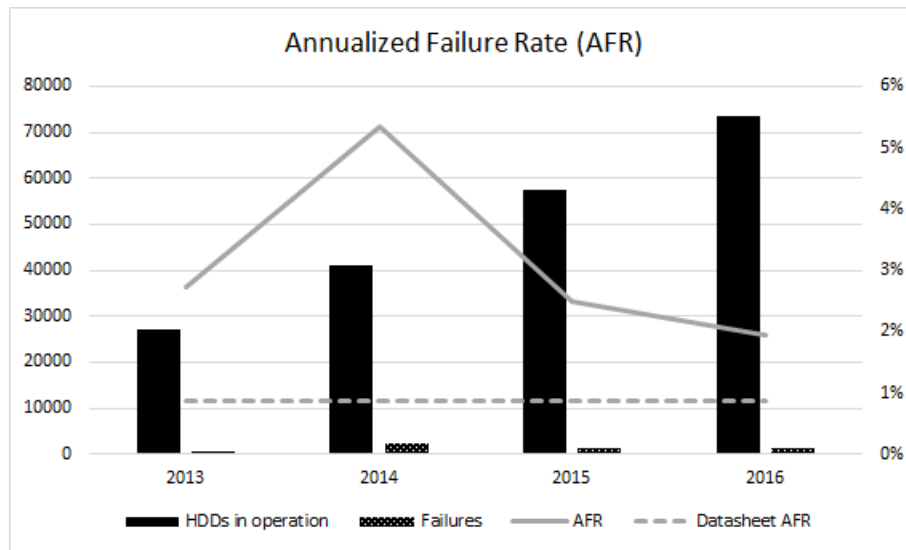


Figure 3. Dataset AFR from 2013 to 2016. In 2014, field-gathered AFR was six times than the highest datasheet AFR informed by manufacturers, which is 0.88% (dashed line).

Each row is related to a single HDD data, and the columns are date, serial number, model, capacity in bytes, failed (boolean value), followed by all SMART parameters available. Figure 4 illustrates rows and columns of the dataset. The serial number uniquely identifies a HDD, and by its model, it was possible to describe the dataset by manufacturer, as shown in Figure 5, in terms of failure events, HDD population, and failure rate. HDD set had models from five major manufacturers.

```
date,serial_number,model,capacity,failure,smart_1_normalized,smart_1_raw,...
2016-12-31,ZA10Q2F7,ST8000DM002,8.00E+12,1,64,134452400,...
2016-12-31,MJ0351YNGA62NA,Hitachi HDS5C3030ALA630,3.00E+12,1,76,22676646,...
```

Figure 4. Illustration of rows and columns of the dataset (excerpt).

Western Digital's HDDs displayed the highest failure rate, 20.23% in 2015-2016, while HGST's, a subsidiary of Western Digital, the lowest failure rate, of 0.5% in the same period. HGST had the second largest HDD population, with 24,545 drives by the end of 2016, and Seagate the largest, with 45,531 drives. The total of failed drives was of 2,857 units. Although Seagate had 2,086 failed HDDs, which corresponds to 73.01% of the total of failures, the failure rate was of 4.58% proportionally to its large drive population.

Once a HDD is declared failed, the serial number is removed from dataset on the next day. For Backblaze, a drive is considered to have failed if any of the three events happens: 1) The drive will not spin up or connect to the OS; 2) The drive will not sync, or stay synced, in a RAID Array; 3) The SMART stats they use show values above their thresholds.

We must emphasize three important notes. First, the dataset does not inform the root cause of failure. It implies that a drive declared failed may have "health" SMART stats if a threshold was not exceeded. Also, it is not possible to establish a correlation between SMART logs and other causes of failure.

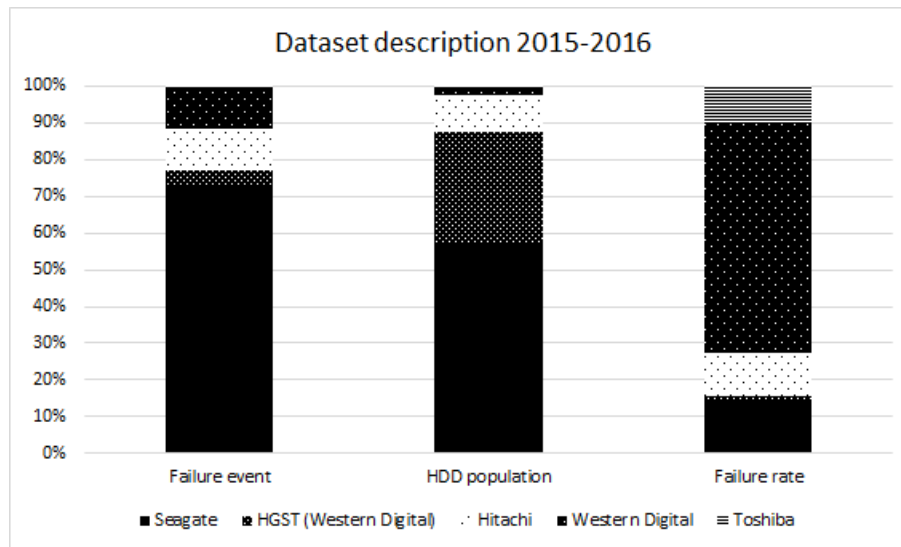


Figure 5. Dataset description from 2015 to 2016 in terms of failure event, HDD population, and failure rate. We highlight the failure rate of Western Digital HDDs, of 20.23%.

Second, Backblaze Inc. keeps track of five SMART attributes: SMART 5 - Re-allocated Sectors Count; SMART 187 - Reported Uncorrectable Errors; SMART 188 - Command Timeout; SMART 197 - Current Pending Sector Count; SMART 198 - Uncorrectable Sector Count.

Hence, if the mentioned conditions *one* and *two* do not happen, only when any of the five SMART attributes overreaches the thresholds established by Backblaze, the drive is classified as failed. Within these conditions, 4.2% of operational drives and 76.7% of failed drives show one or more of those SMART stats with values greater than zero.

And third, SMART-capable devices show both raw and normalized attributes. Raw values are occurrence counters, with the exception of temperature. And normalized values are calculated by manufacturers according to ATA standard. However, not all manufacturers follow this convention.

Our last dataset analysis refers to the number of non-zero SMART stats per failed HDD, as shown in Figure 6. We identified 58 models with four to 21 non-zero parameters. Three out of 2,860 (0.001%) failed drives showed all SMART stats equal to “Null”, or “Not a Number” (NaN). By convention, and in order to process training and prediction algorithms, “Null” and “NaN” values are filled with zeros.

“Null” and “NaN” SMART parameters may happen because of three reasons. First, if a device was not SMART-capable. Second, if any problem blocked log extraction. Or third, if a manufacturer hid data following internal policies.

The very low (0.001%) occurrence of all-zero SMART stats allowed us to review Pinheiro’s conclusion about SMART data effectiveness on HDD failure prediction [Pinheiro et al. 2007]. The difference between the dataset of 2013 and 2016 in terms of data completeness is notable. Extending that to 2005, when Pinheiro collected data for their study, we concluded that the quality of data increased considerably, allowing us to run tests on SMART data to predict HDD failure. Figure 7 shows the number of drives

with four to 21 non-zero SMART stats. There is a consistent amount of useful SMART parameters to be used in this experiment.

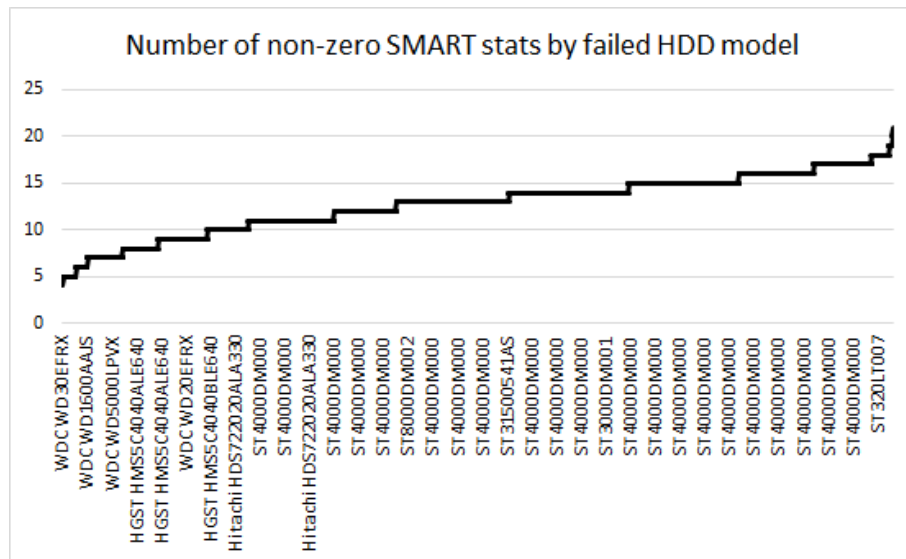


Figure 6. Models of failed HDDs and number of non-zero SMART stats. Not all models are listed due to figure size limits.

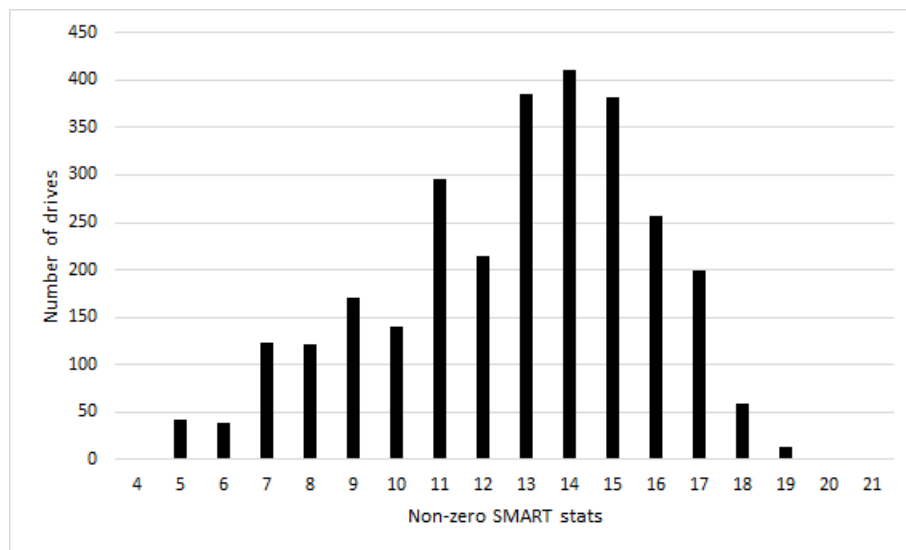


Figure 7. Number of drives per number of non-zero SMART parameters.

4. Experimental Evaluation

Now, we will describe the modeling of data preprocessing and machine learning algorithms. We developed all applications using Python 2.7, and Python scikit-learn 0.18.1. The computer used to run all applications was equipped with a Intel Core i7 3.4GHz processor, 32GB RAM, Ubuntu 16.04 LTS, Linux Kernel 4.8.0. The source code used in our experiments is public available ².

²<http://lnk.ufms.br/smart>

Firstly, we preprocessed the original dataset, extracting all failure instances, and excluding a total of three occurrences because they had all SMART stats equal to “Null”, or “Not a Number” (NaN), as explained before.

Next, as we had an imbalanced dataset, we have chosen a random undersampling approach [Chawla 2005], we concatenated to the failure instances the same amount of non-failing examples from 10 different files, selected in a pseudo-random process, so executing the next steps 10 times to calculate the mean values of accuracy rate and time spent in training and prediction phases.

Then, we split the data in ten train/test sets, using Stratified K-Folds cross-validation. We compared the performance of nine different classification algorithms: Decision Tree (DT), SVM (with linear and RBF kernels), KNN, Multi Layer Perceptron (MLP), Naïve Bayes (NB), Multinomial Naïve Bayes (MNB), Bernoulli Naïve Bayes (BNB) and Logistic Regression (LR).

We carried out a parameter tuning for the first four algorithms through a cross-validated randomized search. The compared values can be observed in Table 2. In the end, considering the best parameters achieved, we gathered the average of ten executions of ten folds each one, that is, the average of 100 different runs for each algorithm, and its respective standard deviation.

Table 2. Parameters tested on tuning procedure.

Algorithm	Parameters	Tested values
SVM	Kernel	Radial Basis Function (RBF) and Linear
	C	Randomly in an exponential function with scale 100.
	gamma	Randomly in an exponential function with scale 0.1.
MLP	solver	SGD (stochastic gradient descent)
	learning_rate	Constant
	momentum	Randomly in an exponential function with scale 0.1.
	alpha	Randomly in an exponential function with scale 0.001.
	activation	Logistic
	learning_rate_init	Randomly in an exponential function with scale 0.01.
	hidden_layer_sizes	(200,50,20), (100,20), (100,50), (200,20), (200,50), (50,10), (200,), (100,), (50,)
max_iter	500	
Decision Tree	criterion	Gini Index
	max_depth	[3,15]
KNN	metric	Euclidean distance
	n_neighbors	[1,15]

Finally, we prepared a model persistence of the best estimator according to its performances and purposes, for future use without having to retrain the model. The fastest algorithms are suitable for short or online tests. The most accurate ones for long or offline

tests.

We also developed an application that extracts SMART parameters from HDDs with exactly same format as Backblaze's does, to run our tests with some diverse data. It should work for both Linux and Windows operating systems with Python 2.7, `smartctl` 6.6, `pySMART` 0.3, and other dependencies installed. `pySMART` is a Python wrapper for the `smartctl` component of `smartmontools`, requesting administrative privileges to access all SMART parameters correctly.

The results are shown in Table 3 and Figure 8. In particular, we have special interest in results in terms of accuracy, which is equivalent to detection rate, and precision, which is complementary to false alarm rate.

It is possible to observe that the decision tree algorithm achieved better results in most of the measures. However, there is no significant difference in terms of performance when comparing decision tree to SVM with RBF kernel. Both algorithms performed very well especially in terms of accuracy and precision. However, decision trees have presented a very competitive performance in terms of execution time. The training and testing procedure in SVM takes much longer time than in decision trees. These results allow us to use decision trees in both online and offline applications to predict disk failures.

As the dataset we used was up-to-date, and more complex than Murray's in terms of disk population, heterogeneity, and completeness, our results show advantages on HDD failure prediction. Also, the persistent models achieved close-to-zero execution time, ensuring performance for both online and offline tests.

Table 3. Performance comparison of machine learning algorithms on training and testing procedures.

Algorithm	Accuracy	Precision	Recall	F1	AUC	Time (sec)
DT	0.80±0.06	0.88±0.03	0.70±0.13	0.78±0.09	0.80±0.06	0.02±0.01
MLP	0.71±0.04	0.72±0.06	0.70±0.10	0.70±0.05	0.71±0.04	6.74±4.84
KNN	0.76±0.07	0.82±0.04	0.66±0.14	0.73±0.10	0.76±0.07	0.05±0.01
NB	0.55±0.03	0.77±0.13	0.14±0.07	0.22±0.11	0.55±0.03	0.01±0.01
MNB	0.61±0.04	0.58±0.03	0.79±0.09	0.67±0.04	0.61±0.04	0.01±0.01
BNB	0.59±0.05	0.57±0.04	0.72±0.09	0.64±0.05	0.59±0.05	0.01±0.00
LR	0.75±0.04	0.81±0.03	0.66±0.07	0.73±0.05	0.75±0.04	0.37±0.10
SVM RBF	0.79±0.06	0.87±0.03	0.69±0.12	0.76±0.09	0.79±0.06	1.84±0.18
SVM Linear	0.78±0.07	0.80±0.05	0.75±0.14	0.76±0.10	0.78±0.07	0.85±0.11

5. Conclusion

This paper has three main contributions. First, SMART data is suitable to solve the problem of predicting HDD failures. A previous work was opposed to this statement because the quality of SMART data in terms of non-zero stats did not allow acceptable results. However, we have identified some remarkable data quality improvements, from 36% all-zero stats down to only 0.001%.

Second, despite of these improvements, not all manufacturers follow ATA standards, probably because of their internal policies. The consequence is that models are

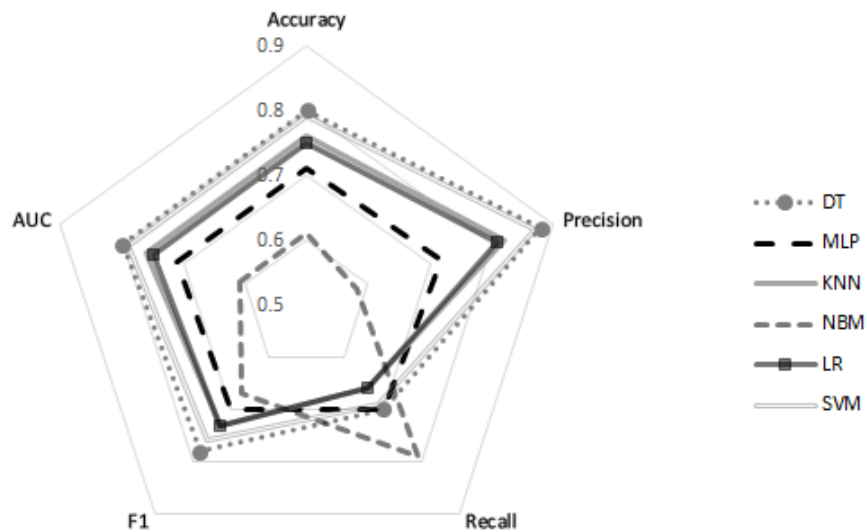


Figure 8. Performance of selected algorithms. We highlight results of decision tree (DT) and Support Vector Machine (SVM).

not totally comparable to each other. Even though, we were capable of summarizing the dataset, and apply the machine learning techniques to detect failures.

Finally, we were able to achieve similar results with decision tree and SVM with RBF kernel algorithms, with around 80% accuracy, 88% precision rates, and 12% in false positive rate in failure predictions. Nevertheless, decision tree was 92 times faster than SVM, fitting both online and offline implementations. Although these rates are lower than those reached in other studies, we presented a more complex problem, with an up-to-date dataset composed of millions of entries from five major HDD manufacturers and 55 different drive models.

Some future works that may enhance our results include improvements on parameter tuning of algorithms, on feature selection of the best SMART stats, and on data clustering. Another approach is to observe new technologies like SSDs (Solid-State Drive), and how to use similar strategies to detect failure on these devices. Finally, we would also consider to separate the dataset by disk manufactures in order to build specialized classifiers to improve failure detection.

References

- Chawla, N. V. (2005). *Data Mining for Imbalanced Datasets: An Overview*. Springer US, Boston, MA.
- Cisco Inc. (2016). Cisco Global Cloud Index: Forecast and Methodology 2015–2020, White Paper. Online; accessed 2017-02-15.
- Dell inc. (2016). Data Loss: Understanding the Causes and Costs, white paper. Online; accessed 2017-02-10.
- Klein, A. (2016). Data Backup: Are You a Hero or a Zero? Online; accessed 2017-01-30.

- Kroll Ontrack Inc. (2010). Lost Data? Reasons and Costs of Data Loss. Online; accessed 2017-03-25.
- Kroll Ontrack Inc. (2014). Hard Disk Drive (HDD) Crashes Remain the Leading Cause of Data Loss, Survey Says. Online; accessed 2017-03-25.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Higher Education.
- Murray, J. F., Hughes, G. F., and Kreutz-Delgado, K. (2005). Machine learning methods for predicting failures in hard drives: A multiple-instance application. *J. Mach. Learn. Res.*, 6:783–816.
- Pinheiro, E., Weber, W.-D., and Barroso, L. A. (2007). Failure trends in a large disk drive population. In *5th USENIX Conference on File and Storage Technologies (FAST 2007)*, pages 17–29.
- Pitakrat, T., van Hoorn, A., and Grunske, L. (2013). A Comparison of Machine Learning Algorithms for Proactive Hard Disk Drive Failure Detection. In *Proceedings of the 4th International ACM Sigsoft Symposium on Architecting Critical Systems, ISARCS '13*, pages 1–10, New York, NY, USA. ACM.
- Schroeder, B. and Gibson, G. A. (2007). Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You? In *Proceedings of the 5th USENIX Conference on File and Storage Technologies, FAST '07*, Berkeley, CA, USA. USENIX Association.
- Yang, J. and Sun, F.-B. (1999). A comprehensive review of hard-disk drive reliability. In *Annual Reliability and Maintainability Symposium. 1999 Proceedings (Cat. No.99CH36283)*, pages 403–409.

Análise de Agrupamentos através da Otimização por Busca em Grupo Melhorada por Poda

Luciano D. S. Pacífico¹

¹Departamento de Estatística e Informática
(DEInfo) – Universidade Federal Rural de Pernambuco (UFRPE)
Rua Dom Manuel de Medeiros, s/n, Dois Irmãos,
52.171-900 – Recife – PE – Brazil

ldsp.ufrpe@gmail.com, luciano.pacifico@ufrpe.br

Abstract. *Data clustering is an important mechanism for exploratory data analysis. Many clustering algorithms present weak global search capabilities, which may lead to sub-optimal solutions. Evolutionary Algorithms (EAs) are known for their good global search capabilities even when dealing with hard optimization environments. In this paper, an improved Group Search Optimization (GSO) algorithm is proposed for data clustering, which use a weedout operator to improve population diversity. Experiments performed on ten real world benchmark problems show that the proposed algorithm is able to achieve good performances in comparison to other EAs from literature.*

Resumo. *A análise de agrupamentos é um mecanismo importante para a análise exploratória de dados. Muitos algoritmos de agrupamento apresentam operadores de busca global ruins, o que pode levar a soluções sub-ótimas. Algoritmos Evolutivos (EAs) são conhecidos por seus bons operadores de busca global mesmo quando tratando de problemas de otimização difíceis. Neste trabalho, uma variação melhorada da Otimização por Busca em Grupo (GSO) é proposta para a análise de agrupamentos, que faz uso de um operador de poda para melhorar a diversidade populacional. Experimentos executados com dez bases de dados reais mostram que o algoritmo proposto é capaz de obter boas performances em comparação a outros EAs da literatura.*

1. Introdução

A análise de agrupamentos, também conhecida por agrupamento de dados, é uma tarefa de aprendizagem não supervisionada que se caracteriza pela tentativa de separação de um conjunto de observações (*dados*) em categorias (*agrupamentos*), de modo que observações pertencentes ao mesmo agrupamento sejam mais semelhantes entre si que observações pertencentes a agrupamentos diferentes. Na análise de agrupamento, os dados são organizados de acordo com suas características internas, não havendo, desta forma, necessidade de conhecimento prévio sobre os mesmos.

Os algoritmos de agrupamento mais populares sofrem de problemas como a dependência do ponto de partida da busca ou a falta de operadores de busca global eficientes, o que pode levar a problemas como a convergência das soluções finais em regiões sub-ótimas do espaço de busca do problema tratado. Como forma de solucionar essas

limitações, Algoritmos Evolucionários (EAs) têm sido aplicados com sucesso na solução de problemas em análise de agrupamentos.

Os EAs são meta-heurísticas de busca inspiradas na seleção natural das espécies, que executam uma busca minuciosa no espaço do problema tratado através de mecanismos que simulam processos biológicos, como mutação, recombinação e seleção [Eiben and Smith 2010]. Os EAs mantêm um conjunto (*população*) de soluções candidatas (*indivíduos*) do problema a cada iteração (*geração*) de suas buscas, sendo a busca realizada por cada indivíduo da população de um EA guiada com o objetivo de otimizar uma função critério (chamada de função de *fitness*). Essas abordagens são conhecidas por sua capacidade de lidar com ambientes difíceis de otimização e por oferecerem bons operadores para fuga de pontos de ótimos locais, o que garante que esses métodos convergirão em uma tentativa de solucionar o problema em mãos globalmente.

Entre os EAs mais populares estão o Algoritmo Genético (*Genetic Algorithm*, ou GA) [Holland 1992], a Evolução Diferencial (*Differential Evolution*, ou DE) [Storn and Price 1995], a Otimização por Enxame de Partículas (*Particle Swarm Optimization*, ou PSO) [Kennedy and Eberhart 1995], e, mais recentemente, a Otimização por Busca em Grupo (*Group Search Optimization*, ou GSO) [He et al. 2006]. Tais algoritmos variam em relação à forma como seus operadores evolucionários conduzirão a população no espaço de busca do problema, além da inspiração teórica para o desenvolvimento de cada modelo.

Porém, mesmo apresentando bons operadores de busca global, os EAs podem sofrer por problemas decorrentes da perda de diversidade de sua população. Isso ocorre pelo fato de que a maioria dessas estratégias faz uso de operadores de seleção que acabam favorecendo a cópia dos melhores indivíduos da população, mesmo que tais indivíduos se encontrem presos em regiões ótimas locais, o que pode levar à convergência prematura da população.

Neste trabalho, um algoritmo de agrupamento particional é elaborado através de uma versão melhorada do GSO (chamada GSOW), visando a manutenção da diversidade populacional. O GSOW faz uso de um operador alternativo de *producing* para o GSO, assim como o uso de um operador de poda por reinicialização aleatória dos membros mais fracos da população, caso a melhor solução global encontrada não melhore por um número pré-definido de gerações, o que indica que a população pode estar presa em uma região ótima local.

O trabalho está dividido da seguinte forma: o GSO tradicional é introduzido na Seção 2, e em seguida o GSOW é apresentado (Seção 3). Os experimentos são discutidos na Seção 4, e logo após as conclusões do trabalho e algumas tendências de pesquisas futuras são apresentadas (Seção 5).

2. Otimização por Busca em Grupo

O GSO é um EA baseado em uma tentativa de simulação do comportamento de busca de animais sociais, e na teoria dos grupos vivos (*living group theory*). O GSO segue o modelo *Producer-Scrounger* (PS), proposto inicialmente por Barnard e Sibly [Barnard and Sibly 1981], como um *framework* para a análise comportamental das estratégias de busca por recursos empregadas por animais que vivem em grupos.

No GSO, a população G de S indivíduos é chamada de *grupo*, enquanto os indivíduos da população são chamados de *membros*. A busca desempenhada por cada membro no GSO leva em consideração o campo de varredura visual do mesmo. Em um problema definido por um espaço de busca n -dimensional, o i -ésimo membro na t -ésima iteração da busca terá uma posição atual definida pelo vetor $\vec{X}_i^t \in \mathfrak{R}^n$ e um ângulo de cabeça definido pelo vetor $\vec{\alpha}_i^t \in \mathfrak{R}^{n-1}$. A direção da busca do i -ésimo membro, que é um vetor $\vec{D}_i^t(\vec{\alpha}_i^t) = (d_{i1}^t, \dots, d_{in}^t)$, pode ser calculada por $\vec{\alpha}_i^t$ através de uma transformação polar para coordenadas cartesianas dada por:

$$d_{i1}^t = \prod_{q=1}^{n-1} \cos(\alpha_{iq}^t), \quad (1)$$

$$d_{ij}^t = \sin(\alpha_{i(j-1)}^t) \prod_{q=1}^{n-1} \cos(\alpha_{iq}^t) (j = 1, \dots, n-1),$$

$$d_{in}^t = \sin(\alpha_{i(n-1)}^t)$$

Um grupo no GSO consiste de três tipos de membros: *producers*, *scrougers* e *rangers*, sendo os *rangers* uma modificação ao modelo PS original proposta pelo GSO. Durante cada geração do GSO, o membro do grupo que encontrou a melhor reserva de recursos (melhor valor de *fitness*), é escolhido como *producer* [Couzin et al. 2005].

O *producer* executará uma estratégia de varredura baseada em seu campo de visão. Essa estratégia recebe o nome de *producing*. Na t -ésima geração de uma execução do GSO, o *producer* \vec{X}_p^t varrerá o espaço de busca do problema através da observação de três pontos aleatórios em seu campo visual: um ponto a zero grau (Eq. (2)), um ponto no hipercubo à sua direita (eq. (3)), e um ponto no hipercubo à sua esquerda (Eq. (4)).

$$\vec{X}_z = \vec{X}_p^t + r_1 l_{max} \vec{D}_p^t(\vec{\alpha}_p^t) \quad (2)$$

$$\vec{X}_r = \vec{X}_p^t + r_1 l_{max} \vec{D}_p^t(\vec{\alpha}_p^t + \frac{\vec{r}_2 \theta_{max}}{2}) \quad (3)$$

$$\vec{X}_l = \vec{X}_p^t + r_1 l_{max} \vec{D}_p^t(\vec{\alpha}_p^t - \frac{\vec{r}_2 \theta_{max}}{2}) \quad (4)$$

onde $r_1 \in \mathfrak{R}$ é um número aleatório obtido através de uma distribuição normal com média 0 (zero) e desvio padrão 1 (um), $\vec{r}_2 \in \mathfrak{R}^{n-1}$ é uma sequência uniforme obtida aleatoriamente no intervalo (0, 1), $\theta_{max} \in \mathfrak{R}^{n-1}$ é o ângulo máximo de busca e $l_{max} \in \mathfrak{R}$ é a distância máxima de busca, definida pela Eq. (5):

$$l_{max} = \|\vec{U} - \vec{L}\| = \sqrt{\sum_{k=1}^n (U_k - L_k)^2} \quad (5)$$

onde U_k e L_k denotam o limite superior e o limite inferior da k -ésima dimensão do problema, respectivamente.

Se o *producer* for capaz de encontrar uma posição melhor que a atual, ele se dirigirá para esse ponto; caso contrário, o *producer* permanecerá em sua posição atual,

movimentando sua cabeça para uma posição definida por um novo ângulo, de acordo com (Eq. (6)):

$$\vec{\alpha}_p^{t+1} = \vec{\alpha}_p^t + \vec{r}_2 \beta_{max} \quad (6)$$

onde $\beta_{max} \in \mathfrak{R}$ é o ângulo máximo de retorno.

Se após $a \in \mathfrak{R}$ gerações o *producer* não for capaz de encontrar uma posição melhor que sua posição atual, ele retornará sua cabeça para a posição de ângulo zero (Eq.(7)).

$$\vec{\alpha}_p^{k+a} = \vec{\alpha}_p^k \quad (7)$$

Todos os *scroungers* tentarão alcançar as reservas já encontradas pelo *producer*, sendo essa estratégia denominada *scrounging*. O operador de *scrounging* no GSO é dado por (Eq. (8)):

$$\vec{X}_i^{t+1} = \vec{X}_i^t + \vec{r}_3 \circ (\vec{X}_p^t - \vec{X}_i^t) \quad (8)$$

onde $\vec{r}_3 \in \mathfrak{R}^n$ é uma sequência uniforme aleatória obtida no intervalo (0, 1), e \circ é o produto de Hadamard ou Schur, que calcula o produto interno entre dois vetores.

Os *rangers* irão executar buscas aleatórias no espaço do problema, sendo essa estratégia denominada *ranging* [Higgins and Strauss 2004]. O operador de *ranging* é apresentado abaixo (Eq. (9)).

$$\vec{X}_i^{t+1} = \vec{X}_i^t + l_i \vec{D}_i^t(\vec{\alpha}_i^{t+1}) \quad (9)$$

onde

$$l_i = ar_1 l_{max} \quad (10)$$

O GSO define uma estratégia de contenção aos membros que, em decorrência da atualização de suas posições pelos operadores evolucionários da estratégia, acabam sendo posicionados fora dos limites do espaço de busca do problema. No GSO tradicional, quando um membro escapa dos limites do espaço de busca após a atualização de sua posição, o mesmo será reconduzido à sua posição anterior dentro dos limites do espaço de busca do problema [Dixon 1959]. A importância do tratamento de indivíduos da população que extrapolam os limites do espaço de busca do problema está relacionada ao fato de que soluções inválidas (que não se enquadrariam às restrições do problema tratado) poderiam ser geradas e consideradas como soluções possíveis para o problema.

O GSO é apresentado no Algoritmo 1.

3. Abordagem Proposta

Nesta seção, o GSOW proposto será apresentado. O GSOW segue o paradigma de algoritmos de agrupamento particionais rígidos, no qual a cada iteração de sua execução cada padrão da base de dados pode estar associado a apenas um único agrupamento por vez.

Considere uma partição P_C de uma base de dados com N_O padrões (cada padrão representado por um vetor $\vec{x}_j \in \mathfrak{R}^m$, onde $j = 1, 2, \dots, N_O$) em C agrupamentos. Cada agrupamento será representado por seu vetor centroide $\vec{g}_c \in \mathfrak{R}^m$ (onde $c = 1, 2, \dots, C$).

Algorithm 1 GSO

 $t \leftarrow 0.$
Inicialize aleatoriamente as posições dos membros $\vec{X}_i^{(0)} \in G.$
Calcule o valor da função de adequação de cada membro $\vec{X}_i^{(0)}.$
enquanto (*as condições de término não forem satisfeitas*) **faça**

 Use o melhor membro do grupo (\vec{X}_p^t) para a execução do *producing*.

 Escolha um percentual dos membros restantes para a execução da estratégia de *scrounging*.

 Ranging: Os membros restantes executarão *ranging* por buscas aleatórias no espaço do problema.

 Calcule o novo valor da função de adequação para cada membro.

 $t := t + 1.$
fim_enquanto
retorne $\vec{X}_p^t.$

Cada membro do grupo do GSOW será representado por um vetor $\vec{X}_i \in \mathbb{R}^n$, onde $n = m \times C$, corresponderá a C centroides, um para cada agrupamento da partição. Desta forma, cada membro do grupo codificará uma partição da base de dados, que representa a solução final oferecida pelo mesmo. Na partição $\vec{X}_i.P_C$, determinada pelo membro \vec{X}_i , os padrões da base de dados estarão associados ao grupo $\vec{X}_i.g_j$, ($j = 1, \dots, C$) ao qual sua distância seja mínima.

No GSOW, a inicialização da população ocorrerá pela escolha aleatória de C padrões da base de dados para compor os vetores centroides iniciais de cada membro do grupo.

Cada membro da população será avaliado de acordo com a função de adequação escolhida, tendo a Soma dos Erros Quadráticos dos Agrupamentos (apresentada na Eq. (11)) sido adotada neste trabalho.

$$WCSS(P_C) = \sum_{c=1}^C \sum_{\forall i \in c} d(\vec{x}_i, \vec{g}_c) \quad (11)$$

O processo geracional do GSOW terá início, sendo cada membro modificado através dos operadores de *producing*, *scrounging* ou *ranging*. No GSOW, um operador alternativo de *producing* é apresentado, de modo que cada nova posição visualizada pelo *producer* terá um de seus centros de agrupamento substituídos através do processo de reinicialização aleatória.

Como uma forma de melhoria da diversidade da população, o GSOW apresenta um operador de poda por reinicialização aleatória. Quando o *producer* é incapaz de encontrar melhores reservas de recursos por *maxCount* gerações seguidas, os membros mais fracos da população (ou seja, os membros com piores valores de *fitness*) são reinicializados aleatoriamente pela escolha de C padrões da base de dados para formar seus novos valores de vetores centroides. A reinicialização aleatória permite que regiões ainda não exploradas do espaço de busca do problema passem a ser visitadas, assim como também evita que o *producer* acabe atraindo todos os membros da população para uma região de pontos ótimos locais do espaço de busca do problema.

O GSOW é apresentado no Algoritmo 2.

Algorithm 2 GSOW

$t \leftarrow 0$.

Inicialização: Para cada membro $\vec{X}_i^{(0)}$, escolha aleatoriamente C padrões da base de dados como seus vetores centroides de agrupamentos iniciais \vec{g}_c . Após isso, associe os demais padrões \vec{x}_i da base de dados ao agrupamento mais próximo.

Calcule o valor inicial da função de adequação de cada um dos membros $\vec{X}_i^{(0)}$.

enquanto (*condições de término não forem satisfeitas*) **faça**

Escolha o melhor membro do grupo como o *producer* \vec{X}_p^t para a iteração atual.

Execute o operador de *producing* (apenas o \vec{X}_p^t). Para cada ponto gerado (\vec{X}_z^t, \vec{X}_r^t e \vec{X}_l^t), substitua um de seus centros de grupos através da escolha aleatória de um padrão da base de dados. Após isso, determine sua partição pela associação dos demais padrões da base de dados ao agrupamento mais próximo.

Escolha um percentual dos membros do grupo (exceto o \vec{X}_p^t) para executar o operador de *scrounging*.

Ranging: Os demais membros irão executar a estratégia de *ranging* através de buscas aleatórias no espaço do problema.

Determine as novas partições representadas por cada membro \vec{X}_i^t pela associação de cada padrão da base de dados ao agrupamento mais próximo.

Calcule o novo valor da função de adequação para cada membro \vec{X}_i^t .

se \vec{X}_p^t não mudou durante *maxCount* gerações **então**

Pode um percentual dos piores membros do grupo através da reinicialização aleatória pela escolha de C padrões da base de dados como seus vetores centroides.

fim_se

$t := t + 1$.

fim_enquanto

retorne \vec{X}_p^t .

4. Resultados Experimentais

Nesta seção, os resultados experimentais são apresentados. Para propósitos de comparação, os seguintes algoritmos de agrupamento particionais baseados em EAs foram escolhidos: o GA, o DE, o PSO e o GSO. As configurações para cada EA são apresentadas na Tabela 1, tendo sido obtidas da literatura: GA e PSO [Abdel-Kader 2010], DE [Das et al. 2008] e GSO [Pacífico and Ludermir 2016].

Os experimentos foram conduzidos através do uso de dez bases de dados reais obtidas através do UCI *Machine Learning Repository* [Asuncion and Newman 2007]: *Diabetes*, *E. Coli*, *Glass*, *Heart*, *Image Segmentation*, *Ionosphere*, *Iris*, *Page Blocks Classification*, *Wine* e *Yeast*. As bases de dados selecionadas são todas categorizadas como Problemas de Classificação (os padrões dessas bases possuem a identificação da classe à qual os mesmos pertencem), apresentando graus de dificuldades variáveis para a tarefa de agrupamento, sendo suas características apresentadas na Tabela 2. Para os testes realizados, o número pretendido de agrupamentos C foi considerado igual ao número de classes presentes em cada base de dados.

Table 1. Parâmetros fixos para cada algoritmo.

Algoritmo	Parâmetro	Valor
Todos os EAs	S	20
	t_{max}	100
GA	Taxa de seleção	50%
	Taxa de <i>crossover</i>	0.8
	Taxa de mutação	0.5
DE	Taxa de <i>crossover</i>	0.9
	F	0.8
PSO	w	0.9 à 0.4
	c_1 e c_2	2.0
GSO e GSOW	θ_{max}	π/a^2
	α_0	$\pi/4$
	β_{max}	$\theta_{max}/2$
	Percentual de <i>scroungers</i>	80%
GSOW	<i>maxCount</i>	10
	Percentual de poda	20%

Table 2. Bases de dados reais obtidas através do UCI Machine Learning Repository.

Base de Dados	Atributos	Classes	Padrões
<i>Diabetes</i>	8	2	768
<i>E. Coli</i>	7	8	336
<i>Glass</i>	9	6	214
<i>Heart</i>	13	2	270
<i>Image Segmentation</i>	18	7	2310
<i>Ionosphere</i>	33	2	351
<i>Iris</i>	4	3	150
<i>Page Blocks Classification</i>	10	5	5473
<i>Wine</i>	13	3	178
<i>Yeast</i>	8	10	1484

Todos os algoritmos foram executados no ambiente MATLAB 7.6. Os algoritmos evolucionários testados foram condicionados de acordo com o esquema adotado para o GSOW (vide Seção 3), com cada indivíduo da população representando C centroides de agrupamento. A função de *fitness* adotada para todos os EAs foi o *WCSS*. Trinta testes independentes foram executados para cada algoritmo em cada uma das bases de dados, e todos os algoritmos evolucionários tiveram como ponto de partida a mesma população inicial em cada um dos testes, sendo tal população obtida pela escolha aleatória de padrões da base de dados em análise para compor os centroides iniciais dos agrupamentos de cada EA.

As métricas de comparação adotadas nos experimentos são o Índice de Rand Corrigido (*CR*) [Hubert and Arabie 1985] e a Soma dos Erros Quadráticos dos Agrupamentos (*WCSS*, dado na Eq.(11)).

O Índice de Rand Corrigido avalia o grau de concordância (similaridade) entre duas partições de uma mesma base de dados. O *CR* fornece sua avaliação baseada na verificação das relações entre pares de padrões em cada uma das partições fornecidas (partição *a priori* já conhecida e partição *a posteriori* obtida pelo algoritmo de agrupamento).

Formalmente, tendo em consideração uma partição $U_R = \{u_1, u_2, \dots, u_R\}$ fornecida por um algoritmo de agrupamento e uma partição fornecida *a priori* $V_C = \{v_1, v_2, \dots, v_C\}$ para a mesma base de dados, o CR é definido pela Eq. (12).

$$CR = \frac{\sum_{i=1}^R \sum_{j=1}^C \binom{n_{ij}}{2} - \binom{n}{2}^{-1} \sum_{i=1}^R \binom{n_i}{2}}{\sum_{i=1}^R \binom{n_i}{2} + \sum_{j=1}^C \binom{n_j}{2} - \binom{n}{2}^{-1} \sum_{i=1}^R \binom{n_i}{2} \sum_{j=1}^C \binom{n_j}{2}} \quad (12)$$

onde n_{ij} representa o número de padrões que estão em ambos os agrupamentos u_i e v_j ; n_i indica o número de padrões no agrupamento u_i ; n_j indica o número de padrões no agrupamento v_j ; e n é o número total de padrões da base de dados estudada. O CR tem seus valores limitados ao intervalo $[-1, 1]$, sendo o valor 1 indicativo de uma perfeita concordância entre as duas partições em comparação, enquanto valores próximos de 0 (ou negativos) um indicativo de partições formadas ao acaso [Arabie et al. 1996].

Os resultados experimentais incluem o valor médio (e desvio padrão) para cada uma das duas métricas de avaliada. O critério de avaliação inclui um sistema de *ranks* obtido através da aplicação de testes de hipóteses do tipo teste de Friedman [Friedman 1937, Friedman 1940] para as métricas adotadas.

O teste de Friedman é um teste de hipóteses não-paramétrico que calcula valores de *ranks* para os algoritmos para cada base de dados separadamente. Se a hipótese nula de que os *ranks* não são significativamente diferentes for rejeitada, o teste de Nemenyi [Nemenyi 1962] é adotado com um teste *post hoc* para o teste de Friedman. De acordo com o teste de Nemenyi, a performance de dois algoritmos é considerada significativamente diferente se a diferença entre seus valores médios de *rank* for ao menos maior que uma *diferença crítica* dada por:

$$CD = q_\alpha \sqrt{\frac{n_{alg}(n_{alg} + 1)}{6n_{bases}}} \quad (13)$$

onde n_{bases} representa o número de bases de dados, n_{alg} representa o número de algoritmos comparados e q_α são valores críticos baseados em estatísticas nos limites do modelo *t* de Student divididas por $\sqrt{2}$ [Demšar 2006]. Nos testes realizados, o grau de significância foi fixado em $\alpha = 0.05$ para o teste de Friedman.

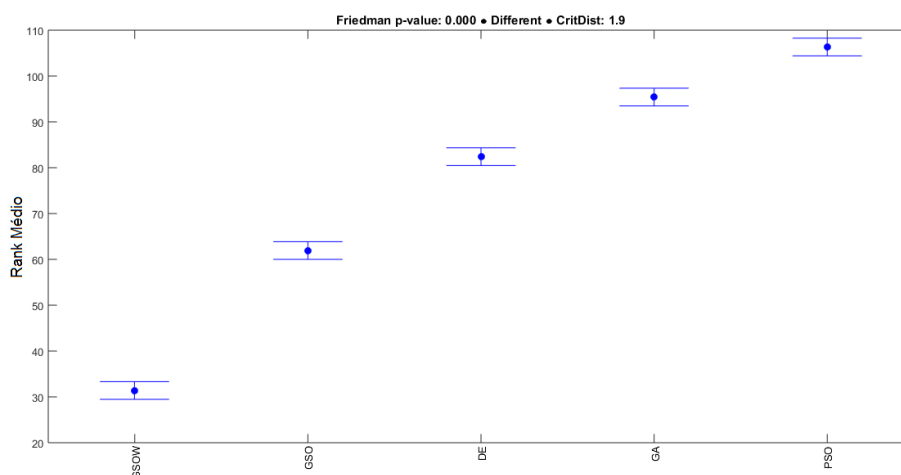
Os resultados experimentais são apresentados na Tabela 3. Para cada base de dados, os melhores valores obtidos para o $WCSS$ e para o CR são apresentados em negrito. Os resultados experimentais mostram que o GSOW proposto foi capaz de obter os melhores valores para o $WCSS$ em oito das dez bases testadas, assim como o melhor valor para o CR para seis dessas bases. O GSOW foi capaz de obter melhores performances que o GSO padrão para nove das dez bases de dados testadas em relação ao $WCSS$, mostrando que o operador de poda por reinicialização aleatória apresentado é capaz de oferecer ao GSO uma boa alternativa para fuga de pontos de ótimos locais, garantindo a diversidade populacional da estratégia. OS experimentos também mostram que os demais EAs testados sofrem de problemas relacionados a convergência prematura, como foi o caso do PSO, responsável pelos piores resultados em relação à função de *fitness* para a maioria

Table 3. Resultados experimentais.

Base de Dados	Algoritmo	WCSS		CR	
		Média	Desvio Padrão	Média	Desvio Padrão
Diabetes	GA	6.3x10 ⁶	4.1x10 ⁵	0.047	0.016
	DE	5.3x10⁶	7.1x10⁴	0.072	0.003
	PSO	6.2x10 ⁶	2.9x10 ⁵	0.049	0.013
	GSO	5.6x10 ⁶	3.1x10 ⁵	0.060	0.012
	GSOW	5.3x10⁶	1.0x10⁵	0.073	0.005
E. Coli	GA	2.965	0.304	0.405	0.079
	DE	20.88	0.911	0.399	0.069
	PSO	17.57	2.065	0.438	0.082
	GSO	17.34	0.717	0.427	0.078
	GSOW	16.87	0.761	0.422	0.065
Glass	GA	581.8	40.3	0.196	0.038
	DE	546.8	33.2	0.232	0.035
	PSO	581.8	40.3	0.196	0.038
	GSO	502.5	33.9	0.234	0.029
	GSOW	426.7	27.1	0.249	0.024
Heart	GA	7.0x10 ⁵	4.0x10 ⁴	0.021	0.017
	DE	5.7x10 ⁵	1.2x10 ⁴	0.031	0.007
	PSO	7.0x10 ⁵	4.0x10 ⁴	0.021	0.017
	GSO	6.2x10 ⁵	4.0x10 ⁴	0.024	0.014
	GSOW	5.6x10⁵	9.7x10³	0.033	0.009
Image Segment.	GA	2.4x10 ⁷	1.6x10 ⁶	0.292	0.048
	DE	2.4x10 ⁷	1.7x10 ⁶	0.272	0.055
	PSO	2.4x10 ⁷	1.6x10 ⁶	0.292	0.048
	GSO	2.3x10 ⁷	1.6x10 ⁶	0.298	0.055
	GSOW	2.1x10⁷	2.3x10⁶	0.303	0.046
Ionosp.	GA	2890.1	202.7	0.084	0.070
	DE	2933.2	209.6	0.081	0.070
	PSO	2702.8	110.2	0.110	0.066
	GSO	2576.6	87.1	0.135	0.051
	GSOW	2480.3	31.3	0.162	0.019
Iris	GA	114.8	11.8	0.724	0.080
	DE	93.98	8.906	0.743	0.043
	PSO	114.8	11.8	0.724	0.080
	GSO	83.61	4.150	0.758	0.030
	GSOW	81.40	1.765	0.748	0.028
Page Blocks Class.	GA	9.3x10 ¹⁰	2.9x10 ¹⁰	0.012	0.032
	DE	3.0x10 ¹⁰	4.7x10 ⁹	0.068	0.030
	PSO	5.9x10 ¹⁰	1.2x10 ¹⁰	0.016	0.034
	GSO	3.6x10 ¹⁰	7.9x10 ⁹	0.004	0.025
	GSOW	2.4x10¹⁰	4.7x10⁹	-0.005	0.007
Wine	GA	2.9x10 ⁶	3.2x10 ⁵	0.366	0.031
	DE	2.4x10⁶	4.2x10⁴	0.369	0.006
	PSO	2.9x10 ⁶	3.2x10 ⁵	0.366	0.031
	GSO	2.4x10⁶	3.1x10⁴	0.370	0.006
	GSOW	2.4x10⁶	8.6x10³	0.371	0.0
Yeast	GA	7.725	0.550	0.121	0.028
	DE	67.09	2.366	0.098	0.021
	PSO	64.48	3.620	0.115	0.024
	GSO	55.59	2.343	0.115	0.025
	GSOW	53.53	2.073	0.117	0.017

Table 4. Ranks médios dos testes de Friedman/Nemenyi para cada medida de avaliação.

Algoritmo	WCSS	CR
GA	95.42	64.81
DE	82.04	78.79
PSO	106.34	68.04
GSO	61.93	78.17
GSOW	31.41	87.70

**Figure 1. Teste de Friedman em relação ao WCSS.**

das bases de dados, ou instabilidade e baixa confiabilidade, como o GA e o DE, que embora tenham conseguido obter boas taxas de convergência em poucos casos, também acabaram apresentando desempenhos médios muito ruins para a maioria das bases de dados testadas.

Em uma análise global oferecida pelo teste de Friedman/Nemenyi (Tabela 4), podemos observar que em média o GSOW obteve melhor desempenho que os demais algoritmos avaliados, sendo as diferenças nos resultados apresentados estatisticamente significativas, de acordo com esse teste de hipóteses (com significância $\alpha = 0.05$ e distância crítica $CD = 1.929$). A Figura 1 e a Figura 2 apresentam os resultados dos *ranks* médios para o teste de Friedman-Nemenyi para o WCSS e para a CR, respectivamente, sendo os algoritmos apresentados do melhor (à esquerda) para o pior (à direita).

5. Conclusões

Neste trabalho, uma variação da Otimização por Busca em Grupo foi apresentada, denominada GSOW, para a análise de agrupamentos particional. O GSOW faz uso de um operador alternativo de *producing*, assim como um operador de poda por reinicialização aleatória para os membros mais fracos da população quando o melhor membro da população não é melhorado por várias gerações. Esse operador tem por objetivo aumentar a diversidade populacional, de modo a evitar que todo o grupo acabe preso em regiões de ótimos locais.

Experimentos foram realizados através de dez bases de dados reais de modo a

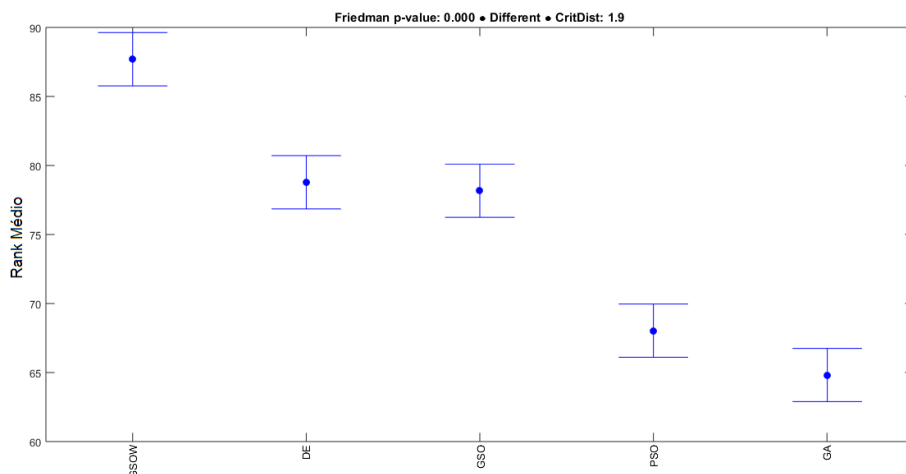


Figure 2. Teste de Friedman em relação ao CR.

comprovar a eficácia do modelo proposto. Para isso, quatro outros algoritmos de agrupamento particionais baseados em EAs foram escolhidos e comparados ao GSOW, sendo eles o GA, o DE, o PSO e o GSO padrão.

Uma avaliação global foi executada através do uso do teste de hipótese de Friedman/Nemenyi. Os resultados experimentais comprovam que o GSOW é uma boa alternativa para a análise de agrupamentos, tendo obtido melhor desempenho médio geral que os algoritmos de teste para as bases de dados adotadas em relação a duas métricas de agrupamento: o *WCSS* e o *CR*.

Como trabalhos futuros, operadores de busca local baseados em algoritmos de agrupamento tradicionais serão adotados para o GSOW com o intuito de acrescentar informações contextuais à busca realizada por este método. Pretende-se ainda adaptar o algoritmo de modo que ele seja capaz de determinar automaticamente o número de agrupamentos naturais apresentado por cada base de dados, eliminando a dependência do fornecimento desse valor como parâmetro.

References

- Abdel-Kader, R. F. (2010). Genetically improved pso algorithm for efficient data clustering. In *Machine Learning and Computing (ICMLC), 2010 Second International Conference on*, pages 71–75. IEEE.
- Arabie, P., Hubert, L. J., and De Soete, G. (1996). *Clustering and classification*. World Scientific.
- Asuncion, A. and Newman, D. (2007). Uci machine learning repository.
- Barnard, C. and Sibly, R. (1981). Producers and scroungers: a general model and its application to captive flocks of house sparrows. *Animal Behaviour*, 29(2):543–550.
- Couzin, I. D., Krause, J., Franks, N. R., and Levin, S. A. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516.
- Das, S., Abraham, A., and Konar, A. (2008). Automatic clustering using an improved differential evolution algorithm. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, 38(1):218–237.

- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30.
- Dixon, A. (1959). An experimental study of the searching behaviour of the predatory coccinellid beetle *adalia decempunctata* (L.). *The Journal of Animal Ecology*, pages 259–281.
- Eiben, A. E. and Smith, J. E. (2010). *Introduction to evolutionary computing*, volume 2. Springer Berlin.
- Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701.
- Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92.
- He, S., Wu, Q., and Saunders, J. (2006). A novel group search optimizer inspired by animal behavioural ecology. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1272–1278. IEEE.
- Higgins, C. L. and Strauss, R. E. (2004). Discrimination and classification of foraging paths produced by search-tactic models. *Behavioral Ecology*, 15(2):248–254.
- Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1):66–72.
- Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE.
- Nemenyi, P. (1962). Distribution-free multiple comparisons. In *Biometrics*, volume 18, page 263. International Biometric Soc.
- Pacifico, L. D. and Ludermir, T. B. (2016). Data clustering using group search optimization with alternative fitness functions. In *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*, pages 301–306. IEEE.
- Storn, R. and Price, K. (1995). Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. international computer science institute, berkeley. Technical report, CA, 1995, Tech. Rep. TR-95-012.

A Comparative Analysis of Neural Networks Implemented with Reduced Numerical Precision and Approximations

Vitor Ferreira Torres¹, Frank Sill Torres²

¹ Graduate Program in Electrical Engineering
Federal University of Minas Gerais
Av. Antonio Carlos 6627, 31270-901,
Belo Horizonte, MG, Brazil

²Department of Electronic Engineering

{vtorres, franksill}@ufmg.br

Abstract. *Machine Learning belongs to a class of computational problems that are resilient to noise and approximations. This is recently being explored in efficient implementations of these algorithms which aim to reduce resource consumption (like memory and energy) while retaining the application performance or trading off some acceptable degradation for the benefits. This paper compares precise and approximated implementations of Artificial Neural Networks, trained with the same architecture and parameters and applied to well known benchmark problems. Results show that direct conversion to more efficient and approximated structures does not always work flawlessly when simplifications are applied not only to the network but also to the training calculations.*

1. Introduction

Machine Learning (ML) applications have become ubiquitous and their implementation is realized in very distinct hardware levels: from embedded platforms, with power, processing and storage limitations to large clusters or tightly coupled parallel processing units. Both levels of hardware platforms are relevant [Wu et al. 2011, Reed and Dongarra 2015] for the ML field and can benefit from optimizations in the final implementations.

Artificial Neural Networks (ANN), an established method to handle ML problems, have been closely related to hardware since their early proposals. ANNs not only were inspired by biological models but were also presented in efficient hardware implementations which became known as “Neuromorphic Computing” [Mead 1990]. These early proposals have not become the standard implementations, largely due to the fast evolution in digital hardware computing performance. The popularity of other methods (e.g. Support Vector Machines and Random Forests) may also have contributed to a reduced interest ANNs in general. After a period without a considerable progress, recent years have seen a few important commercial examples of platforms optimized for the computations used in ANNs: Qualcomm (Zeroth Processor - 2013), IBM [Merolla et al. 2014] (TrueNorth - 2014), NVIDIA (Tesla P100 - 2016), Google [Jouppi et al. 2017] (Tensor Processing Unit - 2016). Intel has also joined this trend [Nervana 2017]. Besides providing tailored architectures for ANNs, a common resource in these hardware solutions is the availability of reduced numerical precision calculations.

Approximate Computing (AC) studies the impact of the lack of hardware reliability, deterministic or not, on applications and its potential for improving efficiency

(e.g., in terms of power consumption or chip area). Related to problems that became a concern when integrated digital circuits reached a level of miniaturization and speed at which reliability started to be an important concern, AC¹ appeared as a technique not only to be aware of these problems but to explore the simplifications to obtain efficiency gains [Agrawal et al. 2016]. Some recent examples of AC used to improve ML efficiency will be presented in Section 2.

The analysis presented in this work intends to compare reference (precise) implementations of ANNs to approximated equivalent networks that require less memory but perform calculations in a less precise manner. Simpler math operations also require less complex hardware or reduced software libraries for low-end processors. This comparison is performed under the same conditions (network structure, activation functions, training algorithm, adjustable parameters and initial weights) to focus the analysis on the approximations themselves.

2. Recent Work

The biological inspiration² of ANNs [Rosenblatt 1958, McCulloch and Pitts 1943] is clearly an intrinsically analog model. The first breakthrough [Williams and Hinton 1986] for their practical implementation provided a way to determine the neuron connection weights by iteratively back-propagating their adjustments from the output to the input layer. These changes were calculated based on the output error of each training example, and applied according the activation derivatives. Although the original model is inherently resilient to noise, this training process must be handle with more care.

Benchmark datasets handled with deep neural networks are tested with three different number representations (floating point, fixed point and dynamic fixed point) and analyzed in [Courbariaux et al. 2014]. Authors come to the conclusion that simple fixed point is considered harmful because activations, parameters and gradients have very different ranges, which may vary during training. The dynamic fixed point implementation was found to be equivalent to the reference floating point, even when used during training.

Energy savings from 34% to 51% in ANNs are reported in [Zhang et al. 2015] by applying approximations in data representation, computation and memory accesses. An iterative optimization heuristics to select candidate neurons and adjust their approximation is presented. The experimental results are obtained by simulation of the real hardware implemented in a 45nm CMOS technology and show less than 5% of quality loss.

In [Gupta et al. 2015] the authors train deep networks with 16 *bits* fixed-point number representations and stochastic rounding. This proposal presents similar results to 32 *bits* floating-point reference implementations in terms of classification accuracy. It is also mentioned that a mixed-precision approach, with higher precision fixed-point operations at the end of the training process, is also a promising strategy.

The paper [Lin and Talathi 2016] reviews recent literature showing that stochastic rounding is a good strategy to improve stability of the training process in deep neural networks under limited numeric precision. Three proposals are presented: low precision weights and full precision activations, fine-tuning only the top layers and a bottom-to-top

¹“deterministic designs that produce imprecise results.” [Han and Orshansky 2013]

²neurons with several inputs and their respective weighted sum, which determines the output activation

iterative fine-tuning scheme which applies approximations progressively. With similar results between the proposals, the best performance was achieved with 8 *bits* weights and 16 *bits* activations, which, if reduced to 8 *bits*, cause only a 2% drop in accuracy.

The analysis presented in [Hashemi et al. 2016] is based on a broad range of numerical representations applied to ANNs in both inputs and network parameters: floating point, fixed point, powers of two multiplications and binary weight nets. The compromise between accuracy and hardware implementation metrics is analyzed and results show a wide range of approximation parameters with negligible degradation in performance.

3. Methodology

The purpose of the following comparison is to perform the training and inference processes of ANNs in different well known datasets and evaluate the performance of approximated implementations. There are many examples in the literature of considerable approximations in the inference phase of the ANNs, but not as many applied during the training process, and even less to the training calculations themselves. Also rare are theoretical analyses of the effect of approximations in each operation on the final result. Another aspect not frequently studied is how to automatically select where and how to approximate, while guaranteeing acceptable results.

The datasets were chosen to allow the use of ANNs with similar topologies: multilayer, fully connected, without recursion or convolution. Even without reaching state-of-the-art results, the focus was not to extract the maximal performance from the baseline, which is the reference implementation with double precision floating point representation and standard math library functions, but to compare its performance with the approximated versions. The type of application was also a factor kept constant, hence all selected benchmarks were classification tasks with supervised training. Data ranges were linearly scaled when needed, using the limits of the training set.

The “Fast Artificial Neural Network Library” (FANN [Nissen 2012]) was used as a basis for the implementation, but was heavily modified. One of the main changes was the integration of part of the Berkeley SoftFloat Library [Hauser 2017], to perform half-precision (16 *bits*) standard [IEEE 2008] floating point operations. All code and modifications were implemented in ANSI C, without architecture specific assembly optimizations, and compiled with GCC 5.4. Compiler optimizations which improved the training time when using the approximated implementations (like aggressive “inlining”) were enabled but no accelerations to the native floating point operations were activated.

Using a standardized floating point format has the advantage to make this investigation relevant for the growing amount of available systems (compilers and hardware) supporting this representation. Data types used in the entire implementation were split in the following groups, selectable at compile time:

- Group 1** variables used for external interaction (dataset loading and saving, ANN persistence, real time statistics etc) fixed in hardware native floating points
- Group 2** variables used directly by the neurons (weights, steepness, neuron inputs and outputs) and the operations involving only these variables (except activation functions and derivative functions, which belong to **Group 3**)
- Group 3** all other variables involved in the forward and backward phases of the ANN execution (including all parameters and operations of the training algorithms)

Analyzing the variable groups that are relevant for the training process, in the **Group 2** were included the most important for memory usage (space and traffic) in the forward phase in large networks and the ones that are frequently used with them (to reduce the need for conversions). Activation functions were not included in this group because their relevance in resource usage decreases with the network size and complexity: the more connections each neuron has, more relevant the sum of products in the inputs will be when compared to the activation. With this reasoning and the irrelevant variables in **Group 1**, the **Group 3** was already defined.

Figure 1 shows graphical representations of two piecewise linear approximations of mathematical functions frequently used in ANN. On the left, the Hyperbolic Tangent ('tanh') reference is compared with the approximated version (implemented in FANN and shown as "approx."), while the relative error is shown in the other 'y' axis. The Logistic function approximation has a very similar aspect since it is only a shifted and scaled version of the Hyperbolic Tangent. In the plot on the right, the exponential approximation [Schraudolph 1999] is also compared with the exact reference, but the difference is barely visible (even if the y axis is shown in logarithmic scale). Relative errors are also presented in the same graph and the observed pattern repeats itself in the whole useful range for 16 bits floating points. This approximation is used for the *Softmax* [Dunne and Campbell 1997] outputs and other activation functions (like the Hyperbolic Tangent and Logistic, when not using the direct piecewise linear versions).

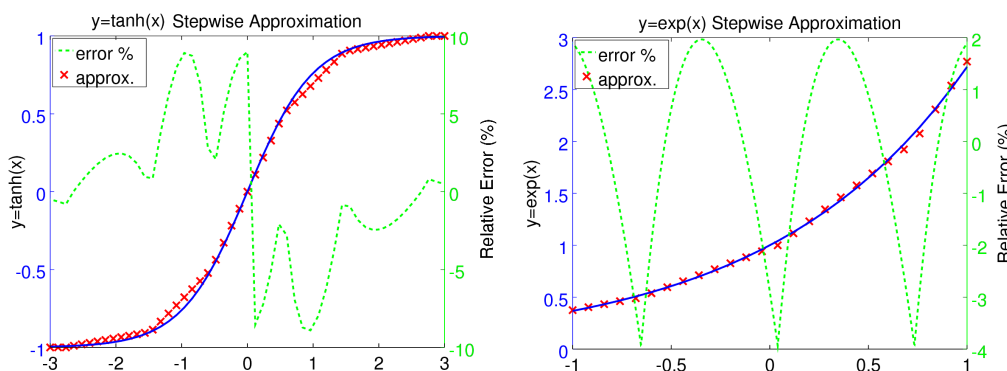


Figure 1. Approximations of Activation Functions and the Exponential

$$\text{Logistic}(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (2)$$

$$\text{Softmax}(\vec{x})_i = \frac{e^{x_i}}{\sum_{k=1}^K e^{x_k}} \quad (3)$$

Equations 1 and 2 show the activation function definitions which were approximated by linear piecewise segments. Equation 3 shows the definition of the *Softmax* function, used for multi-class datasets implemented with mutually exclusive outputs. The actual implementation subtracts the maximum x_i from the exponents for better stability.

Regarding the numerical representation for the variable groups 2 and 3, three fixed implementations were selected for the tests. The baseline (already detailed), an intermediate approximation level (used only with the MNIST database, which will be described later) and the least precise one, used in all databases, as follows:

Group 2 modified half precision floating points, flushing subnormal numbers to zero and saturating maximum numbers (which is a simple modification that saves processing time or circuit/software complexity to handle these exceptions)

Group 3 standard half precision floating points, including the stepwise linear activation and derivative functions and the approximate exponential function for *Softmax* output calculation (using a readily available, but efficient, format for the group that requires more precision)

It should also be noted that conversions from groups 2 and 3 are actually not needed, since their hardware representations (exponent and significant values) are the same. Flushing subnormals to zero and replacing infinities for the maximum value are simple operations.

Weight initialization was also a factor kept constant among all tests and datasets. A normalized [Glorot and Bengio 2010] pseudo-random method was applied to every layer and each random seed was repeated once in the baseline and approximated runs. This assured that, for every baseline execution, there was a respective approximated trial starting from the same random weights (apart from the reduced numerical representations). Since all trials were trained with the same order of examples using the same algorithms and parameters, this procedure allowed a more direct comparison of the approximations effects in the training epochs.

4. Datasets

Table 1 summarizes the main characteristics of the classification datasets selected for this study. There is a mixture of following parameters: set size, input complexity and number of classes (mutually exclusive). Class imbalance also varies and will be analyzed later.

Table 1. Characteristics of the benchmark datasets

Dataset	Inputs	Outputs	Training Set Size	Testing Set Size
MNIST	784	10	60000	10000
Breast Cancer	30	1	455	114
Thyroid	21	3	3600	3600
Soybean	82	19	342	341

For each dataset the initial network architecture was chosen from one literature example and adjusted to be adequate for the intended comparison. This tuning, using the baseline implementation, searched for the highest accuracy on the training dataset (even with overfitting) and just checked if the selected configuration would perform adequately (convergence to a stable result) in the approximated implementation.

In order to obtain ANN configurations with enough capacity to detect possible regularization effects of the approximated methods, the training adjustments have evaluated the training accuracy only with the baseline. To be consistent with this search for an increase in the generalization capability, no regularization technique (e.g. early stop, weight decay, random noise, Dropout) was enabled and, based on the trial runs, a maximum number of epochs was fixed for interruption.

Regarding the training methods, this study preferred standard back-propagation in batch modes (mini-batch for the largest datasets and full-batch for the two others) over adaptive, more robust to parameter changes and faster learning algorithms (like iRPROP [Igel and Hüsken 2000], rmsprop [Tieleman and Hinton 2012] or Adam [Kingma and Ba 2014]). Not even incremental SGD with a momentum factor was thoroughly tested in the approximated implementations, hence not used in the trials. These better training algorithms also add more operations to the basic back-propagation method, which, if not behaved consistently, would also affect the methods based on it.

This methodology for architecture definition and training parameters adjustments resulted in the following configurations:

MNIST two Logistic hidden layers (the first with 50 neurons and the second with 200), *Softmax* output layer, mini-batch training (with batch size = 100).

Breast Cancer two TahH hidden layers (the first with 32 neurons and the second with 8), Logistic output layer, full batch training, data input was linearly scaled to the $[-1.0, 1.0]$ range, using the training dataset limits to calculate the offset and scaling factor, which were applied to the test dataset.

Thyroid two Logistic hidden layers (the first with 40 neurons and the second with 20), *Softmax* output layer and mini-batch training, with a batch size of 50.

Soybean two Logistic hidden layers (the first with 40 neurons and the second with 20), *Softmax* output layer and full batch training.

5. Results and Analysis

The following sections compare the training process of each dataset in approximated implementations to the baseline version, which uses double precision floating points and standard math libraries. Both the training and test accuracy are presented since the latter was not used for any decision during the process.

5.1. MNIST

As mentioned in section 3, the trials with this dataset included an intermediate approximation level. It was used as a first attempt for the approximations, which as then succeeded by the final implementation, already detailed.

This intermediate method was implemented as follows: all data and mathematical operations (groups 2 and 3) were implemented with standard half precision floating point except activation and derivative functions (realized with standard single precision floating points and math library functions). This implementation is documented as “approx. 1” while the other, identical to the one used in the remaining datasets, appears as “approx. 2” in the graphics.

Figure 2 shows in the graph on the left how train and test accuracy evolved on average on all the baseline runs. Approximate trials are not presented in the same graph due to the small differences between them and the baseline. For the same reason, the 95% confidence intervals (0.01% for the last training epoch and 0.05% for the last testing epoch) are not plotted. To make the small performance difference more visible, the graph on the right shows how much better, on average, the approximated versions perform compared to the baseline (a negative value means worse accuracy).

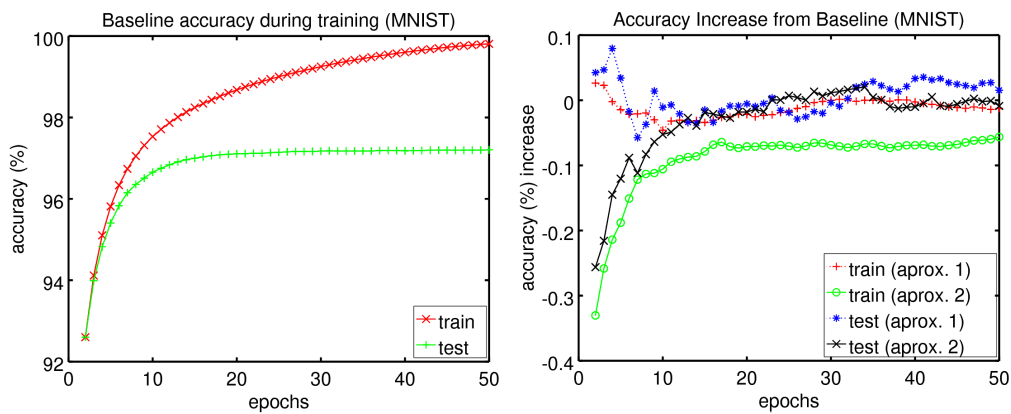


Figure 2. Accuracy Analysis of MNIST dataset

Even with very similar performances observed, the variability of the learning process was analyzed to check if the narrow confidence intervals were hiding less stable results. Figure 3 shows the standard deviation in accuracy along the training process, clearly showing that towards the end of the learning process all implementations converge to similar values of variability among successive independent runs.

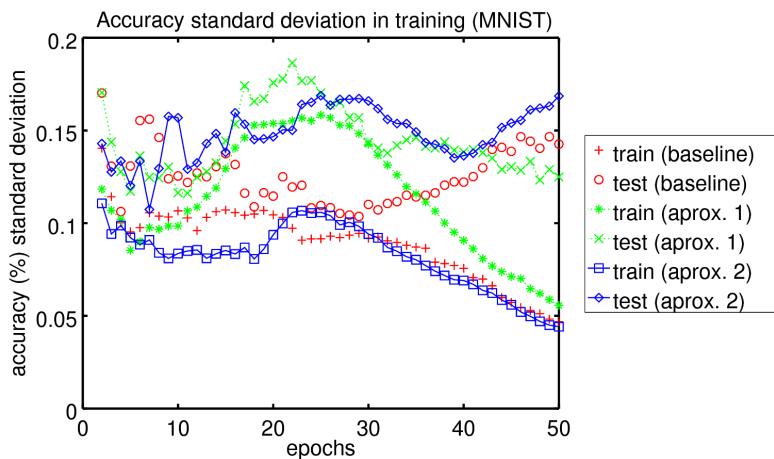


Figure 3. Accuracy Variation during training (MNIST)

Although sometimes considered a “toy” dataset, since the examples are preprocessed in order to reduce the spacial variability found in this type of problems, the presented results allow interesting conclusions. As the ANN proceeded to a clear overfit to the training data, the test performance did not decrease and this behavior was repeated on the approximated versions. Since the test performance difference was not significant, no evidence was found of generalization improvement due do the added noise. This noise was also not enough to make the stability of the convergence considerably worse.

5.2. Breast Cancer (Wisconsin)

This single class classification dataset differs significantly from MNIST. The total amount of examples (569) is two orders of magnitude lower, minimizing the regularization effect found in problems with a large amount of training data. There is also a slight class imbalance: $\approx 41\%$ of the training cases are positive and this number drops to $\approx 23\%$ on the

test set. For this reason, to avoid a benefit to the negative class, the reported accuracy is actually the average of the True Positive Ratio (TPR) and the True Negative Ratio (TNR).

Since the dataset classes are encoded with a single output (with values 0 and 1), the *Softmax* was not used in the output layer. A limited Inverse Hyperbolic Tangent error function was attempted to increase the penalty for larger errors, but the result was not robust in the approximated version. The interesting behavior observed was that in roughly 10% of the trials, after the ANN reached a state of learning stabilization, the weights suddenly saturated (within a single epoch). For this reason, a simple linear error function was used, still achieving better results than with the baseline.

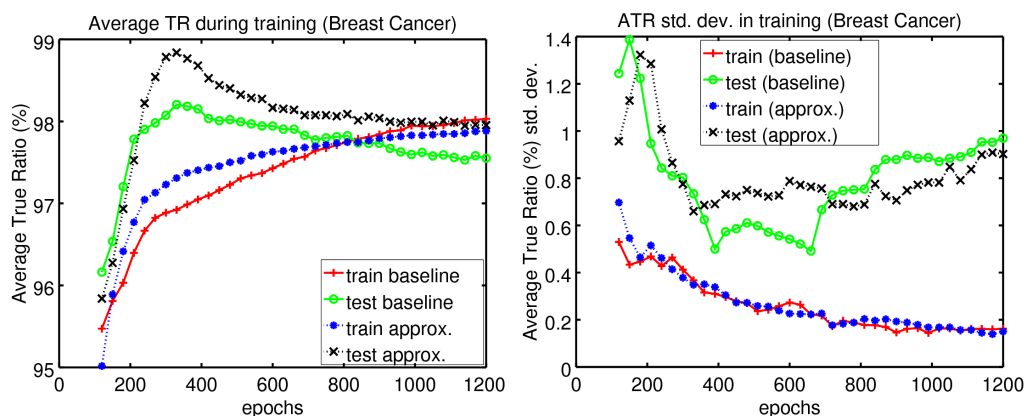


Figure 4. Accuracy Analysis of Breast Cancer dataset

Figure 4 shows in the graph on the left how train and test accuracy evolved on average on all the baseline and approximated runs. A minor but consistent advantage of the approximated version over the baseline is visible. The smallest 95% testing accuracy confidence intervals ($\approx 0.14\%$) coincide with the epoch range where the approximate version reaches its best results. This can be seen in the graph on the right where test performance variability (standard deviation) increases while overfitting proceeds (decreasing also the variability of the training process, which reaches a confidence interval as low as 0.03%). It could be argued whether the small accuracy gain is relevant, specially for a testing dataset where a single positive case missed represents a drop of 0.88% in the TPR, but the consistent differences found encourage further studies on the subject.

5.3. Thyroid

This dataset magnifies a problem found on the Breast Cancer problem: there are three classes but one of them represents more than 92% of the examples. Although the dataset has almost ten times more training data compared to the previous one, this imbalance poses problems to the batch training method used in this study. Overall training accuracy surpasses 99% within the first training epoch, but this number is dominated by the majority class. For this reason, the accuracies reported in the results that follow are the averages of the two minority class accuracies.

The setup chosen for this dataset showed good performance on the baseline but not on the approximated version. It was previously defined in the methodology that no adjustments would be performed to improve performance on the second trial group, otherwise

a direct comparison would not be possible. An instability similar to what was found with the Inverse Hyperbolic Tangent error function in the Breast Cancer dataset was observed.

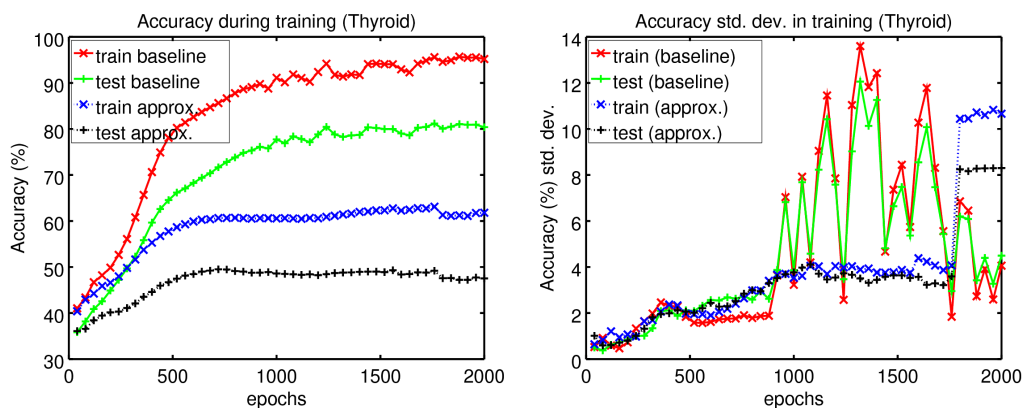


Figure 5. Accuracy Analysis of Thyroid dataset

Figure 5 on the left clearly shows that the accuracy of the approximated implementation is considerable inferior to the baseline. What is not clear is that if the learning rate could be improved in the approximated version, it would show the same variations that appeared in the baseline. The standard deviation plot on the right depicts a smoother approximate training behavior, up to the point when it is affected by the occasional instabilities already mentioned. It is worth pointing out that these issues could easily pass unnoticed if global accuracies had been used. This destructive behavior over the minority classes in the learning process is a known problem in unbalanced datasets.

5.4. Soybean

With this dataset there is a decrease in the set size when compared to the Breast Cancer but the number of classes increases to 19. There is also some class imbalance (the smaller ones have less than 10 examples), but due to the number of classes, the overall accuracy is used in the analysis.

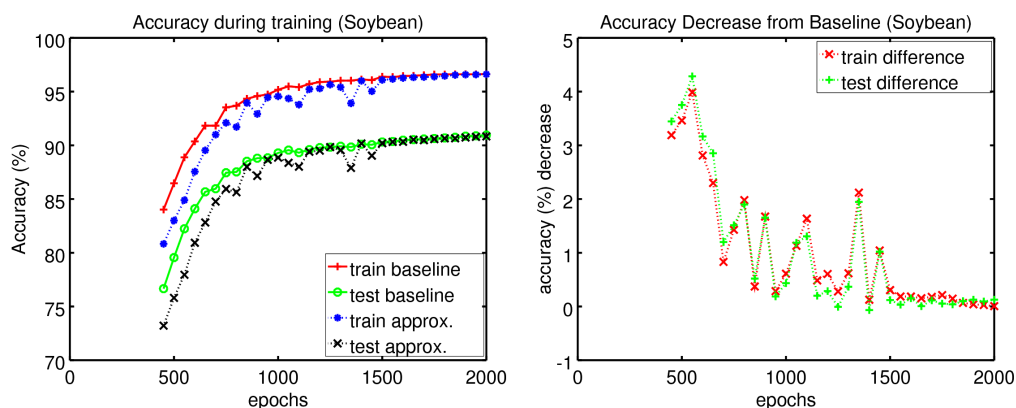


Figure 6. Accuracy Analysis of Soybean dataset

Figure 6 shows in the graph on the left how train and test accuracy evolved on average on all the baseline and approximated runs. The approximated version starts with a minor disadvantage but at the end of the selected training period the difference to the

baseline becomes irrelevant considering the 95% confidence intervals ($\approx 0.21\%$ for testing and $\approx 0.07\%$ for training). The graph on the right depicts how the disadvantage consistently drops towards zero. The pattern is similar to what was observed with the MNIST dataset, without considerable difference from baseline and with no clear overfitting to the training data.

6. Conclusions and Future Work

This comparative study showed some effects that numeric precision and approximations have on the training performance of ANNs. The presented results reinforce the importance of a careful robustness analysis of this ML method when implemented in GPUs (Graphics Processing Units) or other optimized hardware with simpler math operations.

For each dataset a specific analysis of the main results was provided. In two of the examples no evidence was found to show relevant differences between the baseline and approximated versions. In one of them the approximated version behaved consistently better regarding the accuracy and in the other a large disadvantage was observed, although the approximated implementation presented a smoother learning curve.

A next step in this study will be the definition of an architecture selection and parameter optimization method, which would be applied to both implementations using the training data. This could result in different ANNs and training algorithms for the same datasets (or adjustments tuned for the specific problem) and the comparison of the two implementations could show more interesting differences: faster learning times, better resilience to overfitting and a final result with less variability.

Since the main goal of the approximations is the reduction of resources, it would not be acceptable that a complex quality monitoring system had to be added to the system. There is, however, a good opportunity to optimization if a light-weight process could dynamically adjust some approximations and trim down the complexity of the ANN (or even increase it, if the model was found to have insufficient learning capacity).

Finally, an investigation of faster training algorithms could show more vulnerabilities of the approximated approach, demanding more care during the training process to avoid instabilities. These methods will also present new opportunities for approximated arithmetic operations not used in this study. A method that finds superior results when only the best candidates are compared to the ones provided by other method may not be adequate if, on average, the quality of the results is inferior.

References

- Agrawal, A., Choi, J., Gopalakrishnan, K., Gupta, S., Nair, R., Oh, J., Prener, D. A., Shukla, S., Srinivasan, V., and Sura, Z. (2016). Approximate computing: Challenges and opportunities. In *Rebooting Computing (ICRC), IEEE International Conference on*, pages 1–8. IEEE.
- Courbariaux, M., Bengio, Y., and David, J.-P. (2014). Training deep neural networks with low precision multiplications. *arXiv:1412.7024 (Workshop contribution at ICLR 2015)*.

- Dunne, R. A. and Campbell, N. A. (1997). On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne, 181*, volume 185.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256.
- Gupta, S., Agrawal, A., Gopalakrishnan, K., and Narayanan, P. (2015). Deep learning with limited numerical precision. In *ICML*, pages 1737–1746.
- Han, J. and Orshansky, M. (2013). Approximate computing: An emerging paradigm for energy-efficient design. In *Test Symposium (ETS), 2013 18th IEEE European*, pages 1–6. IEEE.
- Hashemi, S., Anthony, N., Tann, H., Bahar, R., and Reda, S. (2016). Understanding the impact of precision quantization on the accuracy and energy of neural networks. *arXiv:1612.03940 (Accepted for conference proceedings in DATE17)*.
- Hauser, J. (2017). Berkeley softfloat. <http://www.jhauser.us/arithmetric/SoftFloat.html>. Accessed: 2017-06-26.
- IEEE (2008). *IEEE Standard for Floating-Point Arithmetic*. IEEE Std 754-2008.
- Igel, C. and Hüsken, M. (2000). Improving the rprop learning algorithm. In *Proceedings of the second international ICSC symposium on neural computation (NC 2000)*, volume 2000, pages 115–121. ICSC Academic Press.
- Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al. (2017). In-datacenter performance analysis of a tensor processing unit. *arXiv:1704.04760 (To appear at the 44th International Symposium on Computer Architecture (ISCA), Toronto, Canada, June 26, 2017.)*.
- Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Lin, D. D. and Talathi, S. S. (2016). Overcoming challenges in fixed point training of deep convolutional networks. *arXiv:1607.02241 (As “Fixed Point Quantization of Deep Convolutional Networks” in Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR)*.
- McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.
- Mead, C. (1990). Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636.
- Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., Jackson, B. L., Imam, N., Guo, C., Nakamura, Y., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673.
- Nervana (2017). Intel nervana hardware. <https://www.intelnervana.com/>. Accessed: 2017-06-26.

- Nissen, S. (2012). Fast artificial neural network library. <http://leenissen.dk/fann/wp>. Accessed: 2017-06-26.
- Reed, D. A. and Dongarra, J. (2015). Exascale computing and big data. *Communications of the ACM*, 58(7):56–68.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.
- Schraudolph, N. N. (1999). A fast, compact approximation of the exponential function. *Neural Computation*, 11(4):853–862.
- Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.
- Williams, D. and Hinton, G. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–538.
- Wu, G., Talwar, S., Johnsson, K., Himayat, N., and Johnson, K. D. (2011). M2m: From mobile to embedded internet. *IEEE Communications Magazine*, 49(4).
- Zhang, Q., Wang, T., Tian, Y., Yuan, F., and Xu, Q. (2015). Approxann: an approximate computing framework for artificial neural network. In *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, pages 701–706. EDA Consortium.

Aprendizagem de máquina para predição de desempenho de estudantes de graduação na UFPE

José O. S. Brandão¹, Adenilton J. Silva¹, Roberta M. M. Gouveia¹, Rodrigo G. F. Soares¹

¹Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco
Rua Dom Manoel de Medeiros, s/n, Dois Irmãos – Recife - PE - Brasil

orlabrandao@gmail.com, adenilton.silva@ufrpe.br,
roberta.gouveia@ufrpe.br, rodrigo.soares@ufrpe.br

Abstract. *This article presents a machine learning approach focused on students with university retention, a problem that concerns the various actors involved with higher education. In this way, a prediction model was developed using the J48, SVM, RBF, Naive Bayes and MLP classifiers individually, and later, to form a committee of these classifiers with Ensemble Bagging and compare the results obtained.*

Resumo. *Este artigo apresenta uma abordagem de aprendizagem de máquinas com foco em estudantes com retenção universitária, um problema que preocupa os diversos atores envolvidos com educação do ensino superior. Dessa forma, foi desenvolvido um modelo de predição utilizando individualmente os classificadores J48, SVM, RBF, Naive Bayes e MLP, para posteriormente, formar um comitê desses classificadores com Ensemble Bagging e comparar os resultados obtidos.*

1. Introdução

Desde a década de 90 que a sociedade brasileira vem demandando por um acesso ao ensino superior mais amplo e inclusivo, pois a economia global está cada vez mais alicerçada no conhecimento. Assim, assistiu-se a um grande aumento na quantidade de jovens e adultos que buscam o ensino superior, fazendo com que o número de estabelecimentos educacionais dessa modalidade de ensino aumentasse proporcionalmente para atender tal demanda. De acordo com o portal do Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) a quantidade de cursos do ensino superior saltou de 4.908 em 1991 para quase 30 mil em 2010, tendo a quantidade de alunos matriculados aumentado de cerca de 1 milhão e meio para quase 6 milhões, no mesmo período, isso apenas dos cursos de graduação presencial.

Essa busca pelo ensino acadêmico foi vital para a ampliação da educação de nível superior no país. Porém, ela veio acompanhada de dois fenômenos que preocupam os profissionais de educação tanto de universidades públicas quanto de universidades privadas, a retenção e a evasão universitárias. A retenção é caracterizada como a permanência do estudante além do tempo máximo de integralização do curso, enquanto a evasão é definida como o abandono do aluno do curso, da universidade ou do próprio ensino superior (Ministério da Educação, 1996).

A retenção e a evasão universitárias vêm sendo estudadas já há algum tempo. A pesquisa realizada por Tinto (1975) é considerada um marco teórico no estudo desses eventos, cujo modelo apresentado sugere que a evasão ocorre pela falta de integração do estudante com o ambiente acadêmico e social da universidade. Neste campo teórico, vários trabalhos mais recentes foram realizados (Ambiel, 2015; Baggi e Lopes, 2011;

Almeida *et al*, 2013). O tema também é bastante estudado por pesquisadores da área de mineração de dados (Romero *et al*, 2008; Dekker *et al*, 2009; Baker *et al*, 2011; Peñayala, 2014).

Com a importância do tema e para ampliar o acesso e a permanência na educação superior, o Ministério da Educação (MEC) criou o Programa de Apoio a Planos de Reestruturação e Expansão das Universidades Federais (REUNI, 2007), tendo como meta global alcançar a taxa de conclusão média dos cursos de graduação presenciais em 90%. Dessa forma, a Universidade Federal de Pernambuco (UFPE) aderiu ao programa do MEC, estipulando como meta reduzir a taxa de evasão de 7,21 para 2% e a taxa de retenção de 58,4% para 20% em 2012 (REUNI/UFPE, 2007). Vale ressaltar que tal meta ainda não foi atingida.

O presente trabalho é um estudo prático e tem como objetivo avaliar a possibilidade do uso de um comitê de classificadores para prever a retenção universitária. Este trabalho estende a pesquisa feita por Hadautho e Adeodato (2012) abrangendo todos os cursos de graduação presenciais da UFPE, do período de 2002 a 2015, utilizando uma abordagem de aprendizagem de máquinas com um comitê (*ensemble bagging*) de classificadores formado por de árvore de decisão, redes neurais, *Support Vector Machine* (SVM), *Radial Basis Functions* (RBF) e *Naive Bayes*, para desenvolver um modelo de classificação de estudantes de graduação com tendência à retenção universitária que venham a atingir um ou vários critérios de jubramento estabelecidos pela universidade. Os dados utilizados nesse estudo foram extraídos do Sistema de Informação e Gerenciamento Acadêmico da UFPE (SIG@ UFPE).

O restante deste trabalho está dividido em seis seções. A seção 2 descreve os trabalhos relacionados. A seção 3 apresenta a principal contribuição deste trabalho, a organização de uma base de dados para o estudo da retenção na UFPE. A seção 4 descreve os modelos de aprendizagem de máquina utilizados neste trabalho. A seção 5 descreve os experimentos realizados, a seção 6 é uma discussão dos resultados obtidos e a seção 7 é a conclusão.

2. Trabalhos relacionados

Esta seção apresenta trabalhos relacionados a evasão e retenção de estudantes. Este é um problema relevante e de importante impacto social e econômico. Márquez-Vera *et al*. (2016) propuseram uma metodologia que consistiu em dividir o semestre em seis etapas, onde cada etapa alimentava uma base de dados com informações sobre os estudantes. Posteriormente, utilizaram um algoritmo específico de classificação para obter um modelo de predição em cada uma das etapas, usando diferentes medidas de desempenho de classificação, com o objetivo de melhorar a precisão do modelo. Tal algoritmo foi chamado de *Interpretable Classification Rule Mining* (ICRM), que utiliza uma variação da Programação Genética, conhecida como *Grammar-Based Genetic Programming* (GBGP).

O trabalho de Maté *et al*. (2016) utilizou dados dos chamados *Massive Open Online Courses* (MOOC), que são os cursos abertos na internet com vários recursos multimídia que incrementam a participação e interação de pessoas de diferentes nacionalidades, como o Coursera ou Udemy. A pesquisa consistiu em aplicar técnicas de mineração de dados e textos, em conjunto com a abordagem de *Learning Analytics* (LA), para incrementar a análise de dados oriundos dos cursos ofertados no ambiente on-line UniMOOC da Universidade de Alicante, na Espanha.

Em sua pesquisa na área de mineração de dados educacionais, Gómez-Rey *et al*.

(2015) faz uma abordagem utilizando regressão ordinal (RO), propondo um modelo gravitacional de dados, do inglês *data gravitational model* (DGM), que aprende os parâmetros utilizando algum tipo de métrica para generalizar problemas de classificação. O algoritmo proposto possui algumas variações, de acordo com a matriz de custo considerada e com a métrica de distância utilizada (Euclidiana ou Mahalanobis), podendo-se utilizar diferentes combinações. Após vários experimentos e comparação com outros algoritmos de classificação nominal como MLP e SVM, ou de técnicas de regressão pura, como Regression Neural Networks (RNN), os pesquisadores chegaram à conclusão que o modelo de regressão ordinal proposto representa “o estado da arte na área de RO”.

Wanli *et al.* (2015) propuseram a construção de um modelo interpretável de predição de desempenho final de estudantes utilizando Programação Genética (PG), *Learning Analytics* (LA), Teoria da Atividade e Mineração de Dados Educacionais (MDE). Eles utilizaram o algoritmo *Interpretable Classification Rule Mining* (GP-ICRM) para generalizar o modelo. Os resultados do GP-ICRM foram comparados com resultados de algoritmos tradicionais de estatística e mineração de dados, como regressão logística, multilayer perceptron e naive bayes, utilizando como medidas de desempenho os valores de *fitness*, precisão e *recall*. Segundo os autores, de um modo geral, o modelo proposto GP-ICRM tem um desempenho superior em torno de 3% em relação aos modelos tradicionais.

Elden *et al.* (2013) propuseram o desenvolvimento de um novo algoritmo chamado Ada-GA, que utiliza AdaBoost *Ensemble* com um algoritmo genético simples para construir um modelo preditivo de classificação de desempenho de estudantes do ensino superior. O Ada-GA é um algoritmo de *boosting* que objetiva otimizar a quantidade de classificadores fracos e seus pesos usando algoritmo genético para tal, com o intuito de melhorar o desempenho do comitê de classificadores. Após os vários experimentos utilizando algoritmos como J48, Naive Update, KMeans para compor o comitê com o AdaBoost, os autores apresentaram uma tabela de comparação de resultados, onde o AdaBoost teve um percentual de precisão preditiva de 81,85% e o modelo proposto apresentou 82,07% de precisão.

A pesquisa de Hadautho e Adeodato (2012) abordou o tema evasão/retenção na Universidade Federal de Pernambuco (UFPE). O estudo foi realizado a partir da base de dados de alunos da UFPE, compreendendo o período de 1998 a 2008 dos cursos de Direito, Ciências Econômicas, Engenharia Civil, Pedagogia, Medicina e Línguas. Eles utilizaram regressão logística e regras de indução para avaliar o risco de evasão/retenção dos estudantes daqueles cursos. Após a aplicação dos algoritmos, observou-se que a base de dados em estudo possuía 47% de estudantes retidos. Para avaliar o risco da solução proposta, Hadautho e Adeodato (2012) utilizaram Kolmogorov-Smirnov (KS-2) e a área sob a curva ROC (AUC_ROC), tendo o desempenho de Max_KS = 0,51 e AUC_ROC = 0,84.

Manhães *et al.* (2011) utilizaram técnicas de mineração de dados para extrair informações de evasão/retenção da base de dados do SIG@ de estudantes dos cursos de graduação da UFRJ, identificando três classes distintas de alunos. A pesquisa avaliou um conjunto de seis algoritmos de mineração de dados do aplicativo WEKA, utilizando a base de dados do SIG@ da UFRJ, foram eles: árvore de decisão (J48 e Simple Chart), vetor de suporte (SVM), estatístico (Naive Bayes), redes neurais artificiais (MLP), *ensemble* (AdaBoost), os quais apresentaram uma acurácia entre 70% a 86%. Ao final eles adotaram o Naive Bayes que, apesar de não ter sido o melhor classificador, apresentou um rendimento global que atendia aos objetivos da pesquisa.

3. Metodologia

A metodologia utilizada seguiu o padrão *Cross Industry Standard for Data Mining* (CRISP-DM) (Wirth e Hipp, 2000). É um processo recursivo, que possui seis etapas: entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação da modelagem e implantação do modelo. Cada etapa deve ser revista até que o modelo apresente resultados satisfatórios, sendo o cientista de dados o profissional que acompanha e executa o processo. Para esta pesquisa, a última etapa de implantação não foi realizada e a ferramenta de mineração de dados utilizada para criação do modelo foi o *framework* WEKA (Witten e Frank, 2017).

A UFPE possui 3 campi nas cidades de Recife, Caruaru e Vitória de Santo Antão. São 12 centros acadêmicos totalizando 100 cursos de graduação regular presencial, dentre outros na modalidade à distância e pós-graduação, com 29.502 estudantes na graduação no segundo semestre de 2014. (Fonte: <http://www.ufpe.br>, acessado em dezembro de 2016)

Os dados utilizados neste estudo foram extraídos do Sistema de Informação e Gerenciamento Acadêmico da UFPE (SIG@ UFPE). A base de dados disponibilizada para este trabalho possuía originalmente 24 planilhas totalizando 3.388.309 registros de dados históricos e 230 variáveis que foram consolidadas através da ferramenta Pentaho Data Integration (PDI). Para esta pesquisa foi considerado o período entre 2002 e 2015, de todos os cursos de graduação da UFPE, exceto medicina. Como exemplo de variáveis originais temos Matrícula, Estado Civil, Nota do ENEM, Código de Disciplina, Nota na Disciplina, Código da Atividade Acadêmica, Nome da Atividade Acadêmica, Componente Curricular, Número de Faltas e Média Final.

Para avaliação do risco de retenção universitária dos alunos, foi considerado como ponto de tomada de decisão (*threshold*) o término do segundo semestre, para que fosse feita a identificação dos alunos que estavam sendo reprovados por falta ou por nota desde o primeiro semestre. A partir da base consolidada com os dados originais, foi realizada uma análise descritiva desses dados. A figura 1 abaixo exibe um gráfico com uma série de reprovações em atividades acadêmicas no período compreendido.

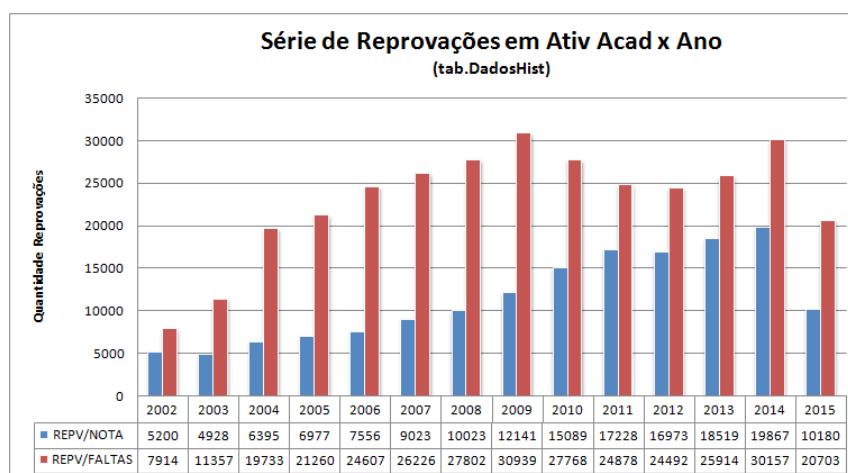


Figura 1. Gráfico com série de reprovações em atividades acadêmicas

O gráfico mostra que há uma crescente quantidade de reprovações desde 2002 em atividades acadêmicas nas quais os alunos são matriculados anualmente. Cabe ressaltar que um aluno pode estar matriculado em uma ou em várias atividades

acadêmicas ao ano, como disciplinas, seminários, estágios, monografia, práticas de ensino, trabalhos de graduação, dentre outras.

A etapa de preparação dos dados, considerada fundamental para o processo de descoberta de conhecimento em mineração de dados, pois os dados originais chegam para o cientista de dados com muita “sujeira”, como dados incompletos, errados, ausentes (*missing data*), desatualizados, inconsistentes, fora do padrão (*outliers*), necessitando serem “limpos” e organizados. Assim, essa etapa consiste do pré-processamento e transformação dos dados originais, deixando-os em condições de serem processados pelos algoritmos de inteligência artificial da ferramenta Weka.

O custo computacional se torna muito elevado para o processamento da quantidade de informações brutas pelos algoritmos de mineração de dados. Dessa forma, convém que seja feita uma seleção/redução dos dados para que, de alguma maneira, não prejudique a eficiência dos métodos de mineração de dados. Assim, alguns critérios de exclusão foram utilizados na base original, como os discentes de cursos EaD, ingressantes extra-vestibular, ingressantes antes de 2002, alunos com vínculos diferenciados, dentre outros.

A outra seleção aplicada aos dados originais foi a redução de dimensionalidade, que consiste na eliminação de atributos, deixando apenas os atributos que são estatisticamente relevantes para o problema em questão. Cabe ressaltar que esta ação foi realizada com o acompanhamento do especialista do negócio, sendo utilizado a técnica *Principal Component Analysis* (PCA). Assim, atributos com altas taxas de *missing datas* foram removidos, bem como as variáveis com dados *a posteriori*, como data de expedição de diploma, ano e semestre de conclusão.

A atividade de transformação dos dados consiste na criação, consolidação, transformação, normalização de atributos com a finalidade de adequar o conjunto de dados para execução dos métodos de mineração de dados escolhidos. Nesta etapa foram geradas novas variáveis para representarem conceitos relacionados ao problema. Como os dados comportamentais se encontravam na granularidade “período letivo / componente curricular / discente” foi necessário fazer uma mudança da base de dados para a granularidade aluno.

Ainda nesta atividade de transformação, foram derivados das variáveis originais, atributos como coeficiente de rendimento, quantidade de aprovações, reprovações, quantidade de faltas, quantidade de aprovações em componentes com alta taxa de reprovação, quantidade de reprovações em componentes com alta taxa de aprovação, dentre outros. Ao final dessa etapa de pré-processamento, o conjunto de dados ficou com 27504 registros no grão aluno, 34 atributos, com a classe binária de alunos jubiláveis e não jubiláveis. A tabela 1 abaixo mostra como ficou o balanceamento das classes por área acadêmica.

Tabela 1. Balanceamento da classe binária Jubiláveis

Área Acadêmica	Alunos	% de Jubiláveis	% de Não Jubiláveis
Área I	11244	28%	72%
Área II	5233	43%	57%
Área III	5242	15%	85%
Área IV	3738	29%	71%
UFPE – Vitória	479	9%	91%

UFPE – Agreste	1568	31%	69%
----------------	------	-----	-----

O atributo-alvo (Jubiláveis) foi criado de acordo com a resolução da UFPE de 2015, que estabelece os quatro critérios de recusa de matrícula para todos os discentes que esgotaram o prazo máximo de conclusão do curso, obtiveram quatro reprovações em um mesmo componente curricular, obtiveram reprovações em todas as componentes curriculares do período letivo por duas vezes e obtiveram por duas vezes o coeficiente de rendimento escolar menor que três.

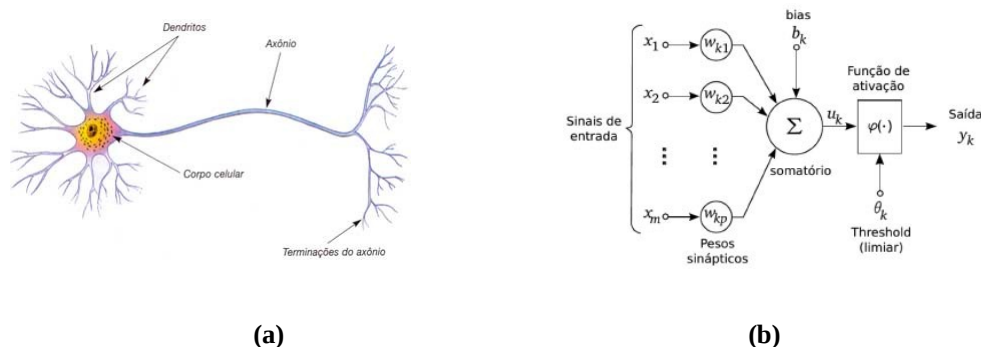
O tópico a seguir apresenta conceitos e definições dos algoritmos utilizados nesta pesquisa. As etapas do CRISP-DM referentes à modelagem e avaliação são referenciadas na Seção 5.

4. Métodos de aprendizagem de máquinas

A abordagem adotada nesta pesquisa consiste em testar seis algoritmos de aprendizagem de máquina (AM): *Naive Bayes*, *Multilayer Perceptron* (MLP), *J48*, *Support Vector Machine* (SVM), *Radial Basis Function* (RBF) e *Bagging*, para verificar suas acurácias na classificação de estudantes com propensão à retenção universitária.

O algoritmo Naive Bayes é baseado no Teorema de Bayes, método probabilístico que conceitua a probabilidade de um evento H dado a evidência E : $\Pr[H|E] = \Pr[E|H]\Pr[H] / \Pr[E]$, onde $\Pr[H]$ é a probabilidade a priori de H e $\Pr[H|E]$ é a probabilidade a posteriori de H . É provavelmente o classificador mais utilizado em aprendizagem de máquina, sendo denominado como ingênuo (*naive*, em inglês) por assumir que os atributos da base a ser trabalhada são condicionalmente independentes, ou seja, não possuem relação de dependência entre eles. Apresenta bons resultados quando se tem disponível um conjunto de treinamento médio ou grande.

Multilayer Perceptron (MLP) são redes neurais artificiais (RNA) que simulam um neurônio biológico por meio de um modelo formal baseado na fisiologia básica desse neurônio. A figura 2 abaixo mostra a comparação de um neurônio biológico e um neurônio artificial *perceptron* usado em algoritmos de RNA.



(a) Fisiologia básica de um neurônio biológico;
(b) Estrutura de um neurônio artificial
Fonte: <https://www.researchgate.net>

É possível fazer um paralelo entre o funcionamento do neurônio artificial *perceptron* e um neurônio biológico: os dendritos, ou cada atributo descritivo $\{x_1, x_2, \dots, x_m\}$ de um exemplar de entrada x_i , estão associados a um neurônio k por meio de seus pesos sinápticos (w_k); o núcleo do corpo celular é representado por uma função

somatória (Σ) que combina as entradas do neurônio; ao resultado desse somatório é aplicada a função de ativação (ϕ), gerando um valor de saída y_k no axônio do neurônio (SILVA *et al*, 2016). As redes neurais podem ser utilizadas em tarefas de classificação, regressão e segmentação.

O J48 é um algoritmo indutor *top-down* de árvore de decisão (AD) muito utilizado em AM. Segundo Quilan (1993), é uma recodificação do algoritmo C45, escrito na linguagem C, sendo considerado o algoritmo que melhor apresenta resultados na montagem de árvore de decisão, utilizando a abordagem de dividir-para-conquistar, decompondo um problema complexo em subproblemas simples, aplicando recursivamente a mesma estratégia a cada subproblema. Possui uma estrutura de fácil manipulação, tendo pouca robustez para trabalhar com bases de grande dimensão e dificuldades em lidar com dados contínuos e ruidosos.

Máquina de vetor de suporte (SVM, do inglês *Support Vector Machine*) constitui uma técnica de AM que divide o espaço de dados linearmente separável em hiperplanos que separam as instâncias de classes diferentes, buscando determinar os vetores de suporte que encontram o hiperplano ótimo, aquele que possui a maior margem e o menor erro de classificação. As SVM são robustas diante de base dados com grande dimensão, com muitos ruídos ou com *outliers*, podendo ser utilizadas tanto em classificação quanto em regressão. Além disso, o uso das funções de kernel na não-linearização das SVM torna o algoritmo muito eficiente (LORENA e CARVALHO, 2007).

As redes RBF (*Radial Basis Function*) são redes neurais multicamadas que podem aprender problemas complexos. Possui uma arquitetura dividida em três camadas: camada de entrada; apenas uma única camada intermediária ou escondida que realiza um mapeamento não-linear, cujos neurônios possuem função de ativação gaussiana, em vez de sigmoideal, como as redes MLP; e a camada de saída linear, que apresenta a resposta da rede ao padrão de ativação aplicado na entrada. Portanto, redes RBF transformam um problema de classificação não-linear em um problema linear.

O algoritmo *Bagging* (*Bootstrap Aggregation*) é um meta-algoritmo para construção de classificadores agregados, que foi proposto por Breiman (1996). Objetiva construir um comitê composto por vários algoritmos classificadores para, então, combinar suas saídas de modo que o desempenho final seja melhor do que o desempenho de um único classificador. É fortemente dependente da diversidade desses classificadores, devendo evitar portanto, subconjuntos idênticos (aqueles que apresentam os mesmos erros) e subconjuntos disjuntos (aqueles cujos erros não se correlacionam). Obtém diversidade através da amostragem de subconjuntos de dados, que são usados para treinar cada classificador do *Ensemble*, cujas saídas são combinadas por meio do voto majoritário.

5. Resultados

A principal ferramenta utilizada neste trabalho foi o *framework* WEKA (*Waikato Environment for Knowledge Analysis*) (Witten e Frank, 2017), que consiste de uma aplicação desenvolvida em Java, contendo uma coleção de algoritmos de aprendizagem de máquinas, desenvolvida por um grupo de pesquisadores da *University of Waikato* na Nova Zelândia.

O conjunto de dados de trabalho possuía 27.504 instâncias, 34 atributos e 2 classes, portanto um problema com rótulo binário. Antes de executar os algoritmos para modelagem, foram utilizadas duas ferramentas do Weka Explorer, o *SpreadSubSample*,

um filtro para instâncias de classificadores supervisionado que faz uma reamostragem aleatória do conjunto de dados. Os parâmetros normalmente configurados são *adjustWeights*, *distributionSpread*, *maxCount* e *randomSeed* (RATNOO, 2013). Esse filtro reduziu o conjunto de dados para 15714 instâncias, fazendo um balanceamento de classes em 50%.

O outro filtro utilizado nesta etapa foi *CfsSubSetEval*, que permite fazer uma seleção de atributos, escolhendo o subconjunto que contém os atributos com a maior capacidade preditiva individual em relação ao atributo-alvo, ou seja, são selecionados os atributos que possuem mais alta correlação com a classe e baixa intercorrelação com outros atributos. Os principais parâmetros configurados são *locallyPredictive*, *numThreads* e *poolSize* (HALL, 1999). O *CfsSubSetEval* retirou 12 atributos do conjunto de dados, deixando-o com 22 atributos.

Em seguida, foi realizada parametrização dos algoritmos e o experimento foi executado com as seguintes configurações dos classificadores: 20 (vinte) árvores de decisão J48, 60 (sessenta) redes neurais artificiais MLP, 24 (vinte e quatro) SVM, 5 (cinco) redes neurais RBF, 1 (um) classificador *Naive Bayes* e o *Ensemble* com *Bagging*. O processo consistiu em rodar cada classificador individualmente com o conjunto de dados, utilizando parametrização personalizada, para então selecionar os que apresentaram os melhores resultados, objetivando compor o *Ensemble* com o meta-classificador *Bagging*.

Para separação dos conjuntos de treinamento e teste foi utilizado *10-fold cross-validation* com apenas uma repetição, o qual realiza uma validação cruzada estratificada com a quantidade de *folds* fornecida. Após o processamento de cada classificador e com o resultado em mãos, foi realizado o teste de hipóteses utilizando o método *paired t-tester (corrected)*, com nível de significância de 5%.

As medidas de desempenho utilizadas foram *Precision*, *Recall* e *F-Measure*. *Precision* é a medida que verifica a razão entre o número de instâncias verdadeiramente positivas/relevantes (jubiláveis) pelo total de instâncias selecionadas pelo classificador. *Recall* refere-se à razão entre o número de instâncias verdadeiramente positivas/relevantes (jubiláveis) pelo total de instâncias positivas/relevantes existentes no conjunto de dados. *F-Measure* é uma medida de desempenho que utiliza uma combinação das outras duas, calculando uma média harmônica ponderada.

A tabela 2 a seguir mostra como ficaram as medidas de desempenho dos classificadores, com seus respectivos desvios-padrão entre parênteses. Em destaque, os melhores resultados de cada medida.

Tabela 2. Desempenho individual dos classificadores

Classificadores	J48	Naive Bayes	MLP	RBF	SVM
<i>Precision</i>	0,72 (0,01)	0,82 (0,01)	0,77 (0,02)	0,76 (0,04)	0,75 (0,03)
<i>Recall</i>	0,73 (0,02)	0,44 (0,02)	0,69 (0,02)	0,63 (0,03)	0,68 (0,02)
<i>F-Measure</i>	0,73 (0,00)	0,57 (0,02)	0,71 (0,01)	0,68 (0,02)	0,71 (0,03)

Com o resultado dos melhores classificadores, pôde-se montar o comitê configurado com o algoritmo *meta.Bagging*, sendo escolhido como classificador base do comitê a árvore de decisão J48, tendo em vista que esse classificador foi o que apresentou o melhor resultado de *Recall*. O restante do comitê foi formado pelos algoritmos de *SVM*, *RBF*, *Naive Bayes* e *MLP*.

A seguir, serão mostradas as cinco primeiras regras extraídas da árvore de

decisão que apresentou o melhor resultado. Cabe ressaltar que os valores apresentados estão normalizados:

- TaxaAprovação2Período $\leq 0,8$ e QuantidadeAprovações2Período $\leq 0,23$ → Jubilável
- NãoFezEnem e CoeficienteRendimentoEscolar2Período $\leq 0,68$ → Jubilável
- QuantidadeAprovações2Período $> 0,23$ e QuantidadeAprovações1Período $\leq 0,4$ → Jubilável
- NãoFezEnem e CoeficienteRendimentoEscolar2Período $> 0,68$ e CargaHorariaTotalCurso $> 0,68$ → Não Jubilável
- QuantidadeAprovações1Período $> 0,5$ e CoeficienteRendimentoEscolar1Período $> 0,66$ → Não Jubilável

A tabela 3 abaixo mostra como ficou o resultado do meta-classificador *Bagging* rodando com 10 (*Bagging 1*), 50 (*Bagging 2*) e 100 (*Bagging 3*) iterações. Bem como, o novo desempenho dos outros classificadores processando os dados em formato de comitê. Entre parênteses o desvio-padrão de cada resultado e, em destaque, os melhores resultados de cada medida de desempenho.

Tabela 3. Resultado dos classificadores em *Ensemble*

Classificadores	Bagging 1	Bagging 2	Bagging 3	SVM	Naive Bayes	RBF	MLP
Precision	0,73 (0,01)	0,73 (0,01)	0,73 (0,01)	0,74 (0,02)	0,82 (0,01)	0,75 (0,02)	0,73 (0,02)
Recall	0,73 (0,02)	0,74 (0,02)	0,74 (0,01)	0,66 (0,03)	0,44 (0,02)	0,63 (0,03)	0,69 (0,02)
F-Measure	0,73 (0,01)	0,73 (0,01)	0,74 (0,01)	0,70 (0,01)	0,57 (0,02)	0,68 (0,02)	0,71 (0,01)

6. Discussão

O objetivo deste trabalho foi desenvolver um modelo de aprendizagem de máquina para classificar estudantes dos cursos de graduação da UFPE com predisposição à retenção universitária, a partir de informações do sistema de gerenciamento acadêmico da universidade, tendo como fronteira de decisão o segundo semestre de cada curso.

O último levantamento feito pela UFPE sobre evasão e retenção foi em 2006/2007 quando a universidade verificou que a taxa de evasão era em média 7,21% e a taxa de retenção era em média 58,4% (REUNI/UFPE, 2007), portanto bastante elevadas. Neste estudo, verificou-se, conforme demonstrado pela tabela 1, que os índices de estudantes em condições de jubramento por área acadêmica são altos, um indicativo de que a retenção universitária continua bastante elevada.

Acredita-se que a universidade deseja identificar o maior número possível de estudantes de graduação com tendência à retenção. Essa informação foi determinante para a escolha das medidas de desempenho utilizadas durante a seleção do algoritmo classificador, ou seja, o que apresentasse o melhor modelo que atendessem àquela necessidade de identificar a maior quantidade possível de estudantes com propensão à retenção universitária.

Neste contexto, podemos verificar conforme a tabela 3 que o classificador *Naive Bayes* destacou-se com uma capacidade de generalização muito boa quanto à medida *Precision*, 82%, o que significa que teve uma muito boa acurácia na classificação de instâncias relevantes (jubiláveis), mas não sabemos quantos falsos positivos (do inglês *False Positive* (FP)) compõem essa seleção. Porém, apresentou um baixo desempenho no *Recall*, apenas 44%, ou seja, do total de instâncias positivas/relevantes que compõem o conjunto de dados, o *Naive* classificou poucas instâncias verdadeiramente relevantes, os chamados verdadeiros positivos (do inglês *True Positive* (TP)).

O classificador J48 teve o segundo melhor desempenho, ou seja, a segunda melhor generalização individualmente, apresentando índices de *Precision* e *Recall* muito próximos, 72% e 73% respectivamente. Como este foi o maior índice de *Recall* entre os cinco classificadores, o algoritmo J48 foi escolhido para ser o classificador base do *Ensemble*, pois o que se almejava não era encontrar o modelo que apresentasse o maior índice de exatidão ou qualidade (*Precision*), e sim, o maior índice de coberturas (*Recall*). Os outros classificadores apresentaram índices muito próximos para essas duas medidas.

Conforme podemos notar na tabela 3, a capacidade de generalização do *Ensemble*, formado pelos seis classificadores em questão, melhorou o índice de *Recall* apresentado individualmente pela árvore de decisão do J48, mas a melhora não foi estatisticamente significativa. Os algoritmos *SVM*, *RBF*, *MLP* e *Naive Bayes* mantiveram os mesmos índices de *Recall* nesta composição. Desta forma, podemos inferir que a árvore de decisão do algoritmo J48 apresentou neste experimento a melhor generalização para classificação de estudantes com tendência à retenção universitária na UFPE.

7. Conclusão e Trabalhos Futuros

Neste trabalho foram apresentados os métodos de aprendizagem de máquina supervisionada *Naive Bayes*, *Support Vector Machine*, *Multilayer Perceptron*, árvore de decisão J48, *Radial Basis Function* e *Ensemble Bagging*, que compõem o *framework* Weka, para trabalhar com a base de dados extraída do sistema de informação e gerenciamento acadêmico da UFPE, dos estudantes de graduação presencial dessa universidade. Cujo objetivo foi apresentar um modelo que melhor generalizasse o problema da retenção universitária.

Foram realizados experimentos individuais com cada um dos classificadores, parametrizações, testes de hipótese e validação usando a ferramenta Weka e, após a escolha da medida de desempenho *Recall*, a qual melhor se adequou ao problema em questão, pode-se inferir que o modelo de árvore de decisão gerada pelo algoritmo J48 generalizou melhor que os outros classificadores, mesmo quando combinados em *Ensemble* com o método *Bagging*.

Para trabalhos futuros sugere-se que não seja removido da base de dados original os estudantes do curso de medicina, pois, como eles apresentavam baixíssimos índices de evasão e retenção, pode ser que essas informações sejam representativas para o atributo-alvo. Assim como, sejam incorporadas a essa base do SIG@ da UFPE informações sócio-econômicas dos estudantes, acreditando-se que tais informações possam contribuir de forma positiva para melhorar a generalização dos classificadores propostos. Outra possibilidade é o uso de métodos evolucionários para a otimização do comitê de classificadores.

Referências

- Almeida, O. C. S.; Abbad, G.; Meneses, P. P. M.; Zerbini, T. 2013. Evasão em cursos a distância: fatores influenciadores. *Revista Brasileira de Orientação Profissional* 14.1.
- Ambiel, R. A. M. 2015. Construção da Escala de Motivos para Evasão do Ensino Superior. *Avaliação Psicológica* 14.1 (2015): 41-52.
- Baggi, C. A. S.; Lopes, D. A. 2011. Evasão e avaliação institucional no ensino superior: uma discussão bibliográfica. *Avaliação: Revista da Avaliação da Educação Superior* 16.2.
- Baker, R. S. I.; Carvalho, A. 2011. Mineração de dados educacionais: Oportunidades para o Brasil. *Brazilian Journal of Computers in Education* 19.02 (2011): 03.
- Breiman L. 1996. Bagging Predictors. *Machine Learning*, v.24, nr 2, p. 123-140.
- Dekker, G.; Mykola P.; Jan V. 2009. Predicting students drop out: A case study. *Educational Data Mining 2009*.
- Elden, A. S.; Moustafa, M. A.; Harb, H. M.; Emara, A. H. 2013. AdaBoost ensemble with simple genetic algorithm for student prediction model. *International Journal of Computer Science & Information Technology*, v. 5.2, p. 73.
- Gómez-rey, P.; Fernández-navarro, F.; Barberà, E. 2015. Ordinal regression by a gravitational model in the field of educational data mining. *Expert Systems*.
- Hadautho, R. B. S; Adeodato, P. J. L. 2012. A Data Mining Approach for Preventing Undergraduate Students Retention.
- Hall, M. A. 1999. Correlation-based feature selection for machine learning.
- Lorena, A.; Carvalho, A. C. P. L. F. 2017. Uma introdução às Support Vector Machines. *Revista de Informática Teórica e Aplicada*, vol. 14, nº 2, p 43-67.
- Manhães, L. M. B.; Cruz, S. M. S.; Costa, R. J. M.; Zavaleta, J.; Zimbrão, G. 2011. Previsão de estudantes com risco de evasão utilizando técnicas de mineração de dados. *Simpósio Brasileiro de Informática na Educação - SBIE*, v. 1, n. 1.
- Márquez-vera, C.; Cano, A.; Romero, C.; Noaman, A. Y. M.; Fardoun, H. M.; Ventura, S. 2016. Early dropout prediction using data mining: a case study with high school students. *Expert Systems with Applications*, v. 33.1, p. 107-124.
- Maté, A.; Gregorio, E.; Trujillo, J. C. J.; Luján-mora, S. 2016. The improvement of analytics in massive open online courses by applying data mining techniques. *Expert Systems*, v. 33.4, p. 374-382.
- Ministério da Educação, 1996. Diplomação, retenção e evasão dos cursos de graduação em instituições de ensino superior públicas. Disponível em <http://www.andifes.org.br/diplomacao-retencao-e-evasao-nos-cursos-de-graduacao-em-instituicoes-de-ensino-superior-publicas>. Acessado em dezembro de julho de 2017.
- Ministério da Educação, 2007. Reestruturação e Expansão das Universidades Federais – Diretivas Gerais (REUNI). Disponível em <http://portal.mec.gov.br/sesu/arquivos/pdf/diretrizesreuni.pdf>. Acessado em junho de 2017.
- Peña-ayala, A. 2014. Educational data mining: A survey and a data mining-based

- analysis of recent works. *Expert Systems with Applications*, v. 41.4, p. 1432-1462.
- Quinlan, J. R. 1993. C4.5: Programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Ratnoo, P. S. 2013. 7 A Comparative Study of Instance Reduction Techniques.
- Romero, C.; Ventura, S. 2008. Data mining algorithms to classify students. *Educational Data Mining 2008*.
- Tinto, V. 1982. Limits of Theory and Practice in Student Attrition. *The Journal of Higher Education*, 53(6), pp. 687–700. Disponível em <http://doi.org/10.2307/1981525>. Acessado em dezembro de 2016.
- Universidade Federal de Pernambuco (UFPE), 2007. Programa de reestruturação e expansão das universidades federais (Projeto REUNI/UFPE). Disponível em <https://www.ufpe.br/proplan/planos-estrategicos>. Acessado em dezembro de 2016.
- Wanli, X.; Rui, G.; Eva, P.; Sean, G. 2015. Participation-based student final performance prediction model through interpretable Genetic Programming: Integrating learning analytics, educational data mining and theory. *Computers in Human Behavior*, v. 47, p.168-181.
- Wirth, R.; Hipp, J. 2000. Crisp-dm: Towards a standard process model for data mining. *In Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, pages 29–39.
- Witten, I. H.; Frank, E. 2017. Data Mining: Pratical Machine Learning Tools and Techniques. Second. San Francisco - CA: [s.n.].

Improving the Performance of Taxi Service Applications using Multi-Agent Systems Techniques

Leonardo Blanger¹, Valmir Junior¹, Clair J. Jevinski,
Alison R. Panisson², Rafael H. Bordini²

¹Universidade Regional Integrada do Alto Uruguai e das Missões (URI)
Erechim – RS – Brasil

²Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Porto Alegre – RS – Brasil

lblanger@hotmail.com, valmirjunior0088@gmail.com, cl.jevinski@hotmail.com
alison.panisson@acad.pucrs.br, rafael.bordini@pucrs.br

Abstract. *Current taxi service applications provide practical mobility solutions for people around the world, allowing fast and efficient taxi allocation by means of a few touches on their smartphones. Efficient taxi allocation also contributes towards sustainability. In this work, we propose a Taxi Service Application based on the multi-agent paradigm. In particular, we propose an architecture for such distributed applications which allows the application to better manage the computational resources and avoid overloading the system components (agents). Our approach explores the benefits that multi-agent systems provide for developing distributed applications. In order to evaluate our approach, we have implemented a prototype of such application in a well-known multi-agent platform. We evaluate our implementation through various experiments. In particular, we present in this work an evaluation for the application response time, varying the number of agents responding to client requests.*

1. Introduction

Taxi service applications have become very popular in the last few years. The need for practical and efficient transportation system has always been present in almost every big city in the world, and taxi service companies are one of the most common transportation services in urban environments. With the increasing popularity of smartphones, as well as the enhancements in the mobile network technologies in the last decades, many taxi service companies evolved in order to better adapt themselves and embrace these new technologies. Despite the large number of small taxi companies and independent drivers in every city, there are a few names that have been receiving much attention recently, such as EasyTaxi¹, 99Taxis², Cabify³, and, probably the most famous example of a transportation service application, Uber⁴, which redefined the standard organisation for this style of business.

In order to be competitive in the scenario of transport applications, a taxi company needs to be able to establish an internal organisation that allows for dynamic assignment

¹<http://www.easytaxi.com/>

²<http://www.99taxis.com/>

³<http://www.cabify.com/>

⁴<https://www.uber.com>

between the currently available taxis and the client requests. In order to keep their customers, companies have to ensure their assignment method maximises not only their profits, but also the client satisfaction with the service [1], in particular in regards to waiting time. In many small companies, this task is frequently performed with the help of human operators [2], and this approach might lead to several problems: (i) incurring additional hiring costs for the company, (ii) ensuring the quality of the service through human decisions, (iii) restricting the scalability of the system, as the company cannot hire and fire operators according to the variations on taxi demand. Further, in a real-world scenario, there are several complicated situations to be handled, such as the possibility of taxis dynamically joining or leaving the system at any moment and in any part of the city, and the difficulty to predict when and where the clients will be requesting the taxis. Besides, the frequency of client requests might drastically change over the time, ranging from a few clients over the night to many more clients than taxis in the rush hours. A system to perform the assignment of taxis to clients needs to handle all such unpredictable behaviour, and still be efficient and scalable both in the number of requests and the number of taxis, being able to respond even in periods of request overload.

In this work, we propose a taxi service application based on the multi-agent paradigm. Multi-agent systems are known to deal with dynamic environments [3], where agents are able to reorganise themselves in order to achieve the best system performance. Therefore, the multi-agent paradigm fits well with the taxi service scenario, as we describe in this work. Furthermore, in this work, we point out several benefits of using multi-agent systems, providing further evidence that multi-agent systems can be used to develop interesting distributed/mobile applications. Our work points towards using multi-agent systems in real-world applications, and further multi-agent system deployment in Industry is still much needed [4]. The main contributions of this work are: (i) we propose an architecture for taxi service applications using multi-agent systems; (ii) we evaluate the proposed architecture by measuring the response time on a simulated version of the problem.

2. Background

In this section, we provide the background of this work. Our main goal here is to provide an overview of the available technologies to implement our approach, including technologies that allow us to implement mobile and distributed systems running multi-agent systems.

2.1. Agent Oriented Programming Language

Among the many agent-oriented programming languages and platforms, such as Jason, Jadex, Jack, AgentFactory, 2APL, GOAL, Golog, and MetateM, as discussed in [5], we choose the Jason platform [6] for our work. Jason extends AgentSpeak(L), an abstract logic-based agent-oriented programming language introduced by Rao [7], which is one of the best-known languages inspired by the BDI (*Beliefs-Desires-Intentions*) architecture, one of the most studied architectures for cognitive agents.

One important feature of the Jason platform [6] for our application is the approach to agent communication. The communication between agents in Jason is done through a pre-defined (internal) action. The most common syntax for this action is:

```
.send(receiver, illocutionary_force, content)
```

where the `receiver` represents the identification (name) of the agent who will receive the message, the `content` represents a literal although at times it can represent a triggering event, a plan or a list of literals or plans, and the `illocutionary_force` represents the speech act performatives that can be used to communicate. There are a number of performatives allowing a rich communication between agents in Jason, which are explained in detail in [6]. In this work, we used a series of them, in order to perform the exchanging of information (beliefs) between the agents. For example, some of the performatives we use in this work are:

- `tell`: is used when the sender wants the receiver to believe the literal in the messages content.
- `achieve`: is used when the sender wants the receiver to try and achieve some goal.
- `askOne`: is used when the sender wants to know if the content of the message is true for the receiver.

Although, in this work, we use only the already existing performatives, new performatives can be defined in Jason, for example, to enable argumentation-based dialogues [8, 9]. Furthermore, Jason allows us to implement new internal actions in Java for various purposes. For example, as an interface to legacy code, business rules, complex mathematical operations, etc. In this work, among others we implemented an internal action to calculate the suitability of individual taxis, i.e., we calculate the best taxi for a request; in this work, we have used simply the distance between the taxi and the client. More refined calculations of suitability will be used in future versions of the application.

Finally, another interesting characteristic of Jason platform is that agents are able to duplicate themselves (or to create other agents) by means of the predefined internal action `.create_agent(agent_name, agent_code)`. Such feature allows us to treat issues related to the scalability of the system, as we will describe in this work.

2.2. Environment and Network

Environments for multi-agent systems are abstractions of real or virtual environments, offering mechanisms for agents to manipulate and perceive the environment in which they are situated. Multi-agent applications developed using Jason, normally use the CArtaGO Platform [10] for creating agent environments, given the most recent multi-agent framework called JaCaMo [11], which combines the Jason Platform [6] for developing agents, the CArtaGO Platform [10] for developing multi-agent environments, and the Moise Model [12] for organisational specification. CArtaGO is a platform based on the *artifact* notion in multi-agent systems. Artifacts are function-oriented computational abstractions which provide services that agents can exploit to support their activities. An artifact makes its functionalities available and exploitable by agents through a set of operations and a set of observable properties [10]. Operations represent computational processes executed inside artifacts, that may be triggered by agents or other artifacts.

In order to develop mobile application based on the multi-agent paradigm, the JaCa-Android Framework⁵ [13] (also referred to as a Middleware) has been developed. It provides an agent-oriented level of abstraction to design and program *smart mobile applications* on top of the Android platform, using a combination of Jason and CArtaGO.

⁵<http://jaca-android.sourceforge.net/>

JaCa-Android allows agents perceiving and manipulating mobile functionalities, for example, agents are able to perceive the GPS position using the `GPSArtifact` [13], as well as calendars, SMS, etc. Also, in regards to network distribution, `CARTAgO` provides a functionality for creating and joining remote workspaces. Such features allows us to implement multi-agent systems, in which agents run in different hosts, and they are able to communicate to each other in order to coordinate their activities towards achieving the system's goals. Examples of mobile applications already developed using such technologies can be found in [14, 15, 16, 17].

3. Taxi Service Application

In this section, we describe the overall taxi application architecture⁶, starting with the agents that compose the system and then giving an overview of the entire system. Also, we discuss the most interesting aspects of the architecture in the remaining subsections, including the dynamic registration of available taxis, the geometric treatment, the scalability of the system, how we treat requests in overlap areas, and so forth.

3.1. Taxi Application Architecture

The taxi service architecture we propose works maintaining a list of available taxis in each region and assigning the best available taxi among them for each client request. In this work, we use the concept of “agency” that are entities in control of the taxis that are within a region. This way, each agency controls a single specific area, and maintains a fleet composed of all the taxis currently inside this area. When a taxi enters a region, it informs its position to all the agencies, and the appropriate agency (i.e., the one responsible for that particular area) adds it to the list of available taxis in its belief base, as it is illustrated in Figure 1(b). The three different types of agents present in our approach are:

Taxis: The taxis are responsible for representing the drivers, and they are able to join the system at any time. Taxi agents run in the mobile devices and communicate through the mobile network; therefore, they do not execute complex processes nor treat complex requests. The main task for taxi agents is to recover the driver coordinates in the city. When the taxi agent starts to execute, it recovers the driver coordinates and after having the coordinates, it broadcasts a message to all agencies (agents with the role of agency) in the system, informing its availability and its position. Whenever an agency assigns such a taxi to a client, the taxi agent creates a goal to guide the driver to the passenger's coordinates, and then to take them to the client's destiny. Further, the taxi agent is able to communicate to its client agent, sending its current position during the journey to the client.

Client: the client agents represent the clients of the taxi service. Each client has its personal client agent running in its mobile device. Therefore, as for the taxi agent, the client agent does not execute any complex process or request; it works essentially collecting information from the user and interacting with them. When a client requests a ride, its agent recovers its current coordinates and asks the client to determine their destination coordinates in the city. After this, it broadcasts a message to all agencies, informing these initial and destination positions, and then the client agent just waits until a taxi has been allocated.

⁶The system architecture used in this work was inspired by some recent multi-agent applications developed in [16, 17].

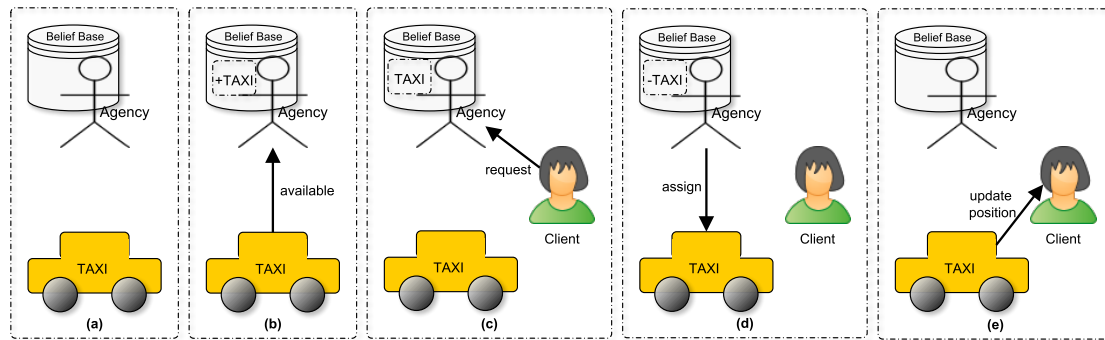


Figure 1. Dynamic Taxi Availability.

Agencies: The agencies implement the main logic of the system; they run on servers and have more processing and communication power, so they are responsible for more complex processing. When the system starts, a number of agencies, corresponding to the number of regions the city has, are created and the agencies remain waiting for taxi registrations and client requests. When the agencies receive a message informing that a taxi is now available, each agency checks if this taxi is within its controlled area, and if it is, a belief representing this fact is added to the agency's belief base, as shown in Figure 1 (a) and (b), representing that this taxi is currently part of this specific agency fleet. The areas where each agency is responsible for taxi availability do not overlap with each other, as shown by the bold lines in Figure 2. Therefore, each taxi will be available only for one agency, depending on its current position. On the other hand, a taxi request from a client may be relevant to various different agency areas close to the client's location, so while areas do not overlap for taxi locations, areas for taxi requests do overlap, as shown by the dashed lines in Figure 2; this improves the way we handle the requests, by allowing taxis that are idle inside a region to handle client requests on neighbouring, more busy regions. Thus, agencies are also responsible for handling client requests, and assigning for each of these clients the best possible available taxi in the system, considering that agency areas for client requests overlap each other.

The dispatch of a taxi to a client works as follows: when a client wants a ride, the client agent broadcasts this request to all agencies, and all agencies that overlap in this client's position (i.e., the areas inside the dashed lines in Figure 2) will handle that request. Each one of these agencies is responsible for selecting the best possible available taxi among their fleets. When an agency finishes the selection process, it informs its best option to all other agencies that are treating such a request. When an agency receives a choice from some other agency, related to a specific client, which is better than its best option, this agency just stops processing the request for that client. When an agency receives the choice from all other agencies that are treating such a request, and it still concludes that its choice is the overall best option, than it dispatches the selected taxi to the client, removing that taxi from its list of available taxis. Finally, the taxi and client agents can directly communicate with each other as in Figure 1 (e), for example to update the taxi's current position.

When the taxi becomes available again, it just informs the system and the responsible agency (which controls the area where the taxi is at that moment) registers the taxi in its belief base as an available option.

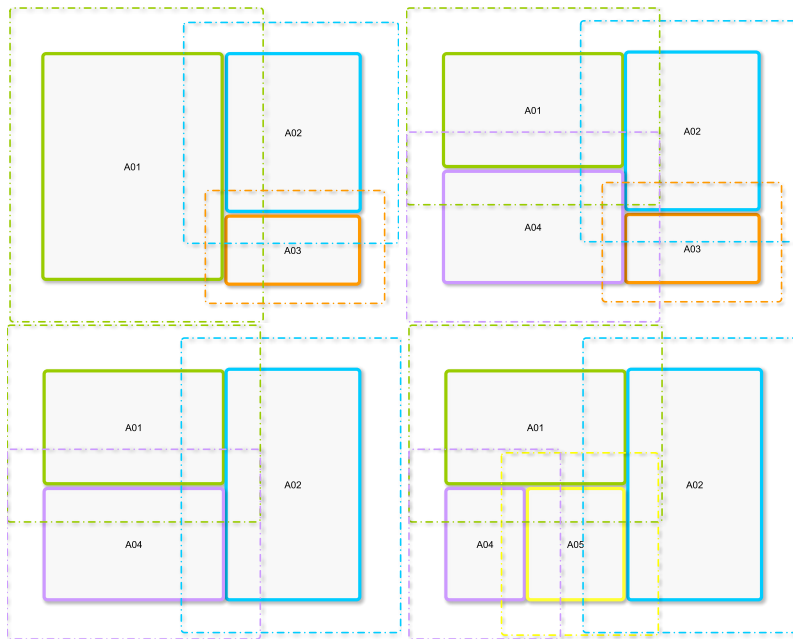


Figure 2. Dynamic reorganisation for geometric treatment. It shows an evolution from the top-left configuration to the bottom-right configuration.

3.2. Dynamic Scalability

The system is able to dynamically reorganise its geometric treatment depending on the number of available taxis and client requests in each area. If an agency agent is currently controlling an area with too many available taxis, or has been receiving too many requests in that area, it will start to have a slower response time. To avoid this problem, whenever the workload of an agency agent exceeds some threshold⁷ (this limit can be applied both to the number of taxis or to passenger requests), then this agency will split itself into two new agencies. Each of this two agencies (the original agency and its duplicate) will divide the respective region and, consequently, each of them will keep a disjoint subset of the taxis from the original agency.

Similarly, if an agency workload is too small (below another predefined threshold), then the agency will search, among its adjacent agencies, for the agency that has the minimum workload, and then “kill” itself, merging its region and respective taxis with this neighbour. This is done to avoid that underused agencies keep executing unnecessarily.

This approach allows the system to maintain the workload balanced among all the existing agencies. Besides, since each agency agent may be run in any server, this approach also allows a better use of distributed computation.

In Figure 2, we also show an evolution of the geometric treatment in our approach. In the top-left image, the system starts with 3 agencies (named A01, A02, and A03) and they reorganise themselves in order to minimise the response time, given a large number of taxis in area A01. Thus, the agency A01 creates a new agency (named A04) and divides its area and the corresponding registered taxis, attributing half of such area for

⁷Such threshold can be either predefined or calculated dynamically based on the response time of the application.

A04. The new configuration is shown in the top-right image. Afterwards, in the bottom-left image, the system reorganises the areas of agencies A02 and A03, where agency A03 passes its area on to agency A02, and agency A02 starts to respond for A03's area as well. Such reorganisation resulted from a low number of taxis and requests in both those areas. Finally, the bottom-right image illustrates the system splitting area A04 because of the overload of taxis and requests in that area, creating area A05 in the process.

3.3. Requests in Overlapping Areas

Our system architecture enables the treatment of client requests by multiples agencies. This occurs every time a client requests a taxi within an overlapping area, i.e., an area where multiple agencies treat client requests. An abstract example is shown in Figure 3, where a client starts requesting a taxi in an area treated by agencies *agency1*, *agency2* and *agency3*, represented by the dashed lines in 3(a). All agencies that are responsible for handling requests in that area will receive that client's request. After receiving the client's request, each agency searches for the best taxi available in its taxi area, represented by the coloured blocks, and sends the best *local* option to all other agencies responsible for the same area, as shown in (b). After receiving the best option from all other agencies, the agency that wins the allocation, i.e., the agency that has the best *global* option, sends a message to the selected taxi, as shown in (c). Finally, the taxi and the client can communicate with each other, updating the taxi's position for example, as in (d).

The reason for this approach with overlapping areas for client requests is that if the client request was treated only by a single agency, it is possible that the chosen taxi is not the current global optimal for this client. For example, suppose the case where the requests are handled by a single agency, a client is located on a border of the agency's region, and the only available taxi is currently located on an opposite border. In this situation, it might be advantageous to consider also the taxis that belong to the neighbouring regions, as it is done in our approach.

3.4. Characteristics

In this section, we describe some of the main advantages and characteristics of the proposed approach.

Scalability: The system allows us a flexible and automated expansion in the number of agencies and consequently on the number of taxis and clients handled efficiently. Recall that the drivers are able to join the system at any time and they can exist in the system in any number, and there can be any number of clients requesting or waiting for the service at any moment and in any location of the city. The system is capable of handling a wide range of situations that can happened in a big city. Also, as an agency is allowed to replicate itself in order to dispatch some of its workload to another agent, possibly running in another server, the system will continue to be efficient even in the case of an overload of requests concentrated on a single region.

One of the main advantages of our approach is that it avoids **single points of failure** in the system. The agents that represent the clients and the taxis have a temporary existence in the system, and they do not participate on any decision process. The agency agents, where decisions are actually made, can exist in any predefined number at the

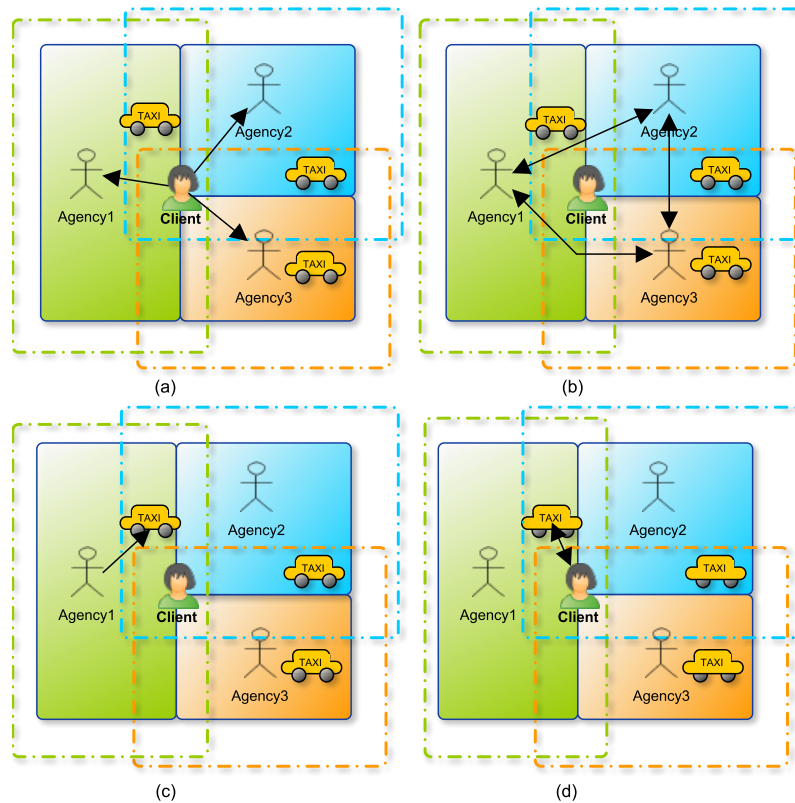


Figure 3. Protocol for Requests in Overlapping Areas.

beginning of the execution, and can split themselves or even quit running in order to better attend the distribution of taxis and clients.

Furthermore, if an agency becomes offline, due to a possible failure on its server, the multi-agent system will be able to identify this situation and instantiate a new agency to respond for the one that is down. This verification can be easily implemented with periodic messages between some of the agency agents. With the help of a redundant implementation for the agencies, the agent running on the new server can have the same state of beliefs that the agent in the failed server had, taking over the control of its taxis and its region, in a completely transparent way for the clients and drivers. This makes the system *tolerant to infrastructure failures*, and prevents it from staying down because of a single agency failure. Besides, as the agents are capable of moving from one host to another, it is possible to apply a load balancing algorithm to improve the performance of the system.

4. Evaluation

In order to evaluate the current prototype of our taxi service application, we have run some experiments. One of the most interesting experiments we have run so far is related to the treatment of one client request by multiple agents, regarding our approach introduced in Section 3.3. In this experiment, a client agent requests a taxi and the agencies (or agency) responsible for client requests in that area proceed as described earlier, finding the best local option and the best global option afterwards. In this initial setting, we used 100000 available taxis with fixed positions, and we distribute such available taxis varying the

number of agencies. First, we start with only one agency treating the client requests, and it has to find the best option, given the 100000 available taxis in its belief base. After that, we split such 100000 available taxis in 2 agencies, where both of them treat the client requests. After that we continue splitting the 100000 available taxis in 3, 4, 5, 10, 20, 30, 40, and 50 agencies, where in all cases all agencies treat the same client requests, find the best local option and use the protocol to find the best global one. The results are presented in Figure 4.

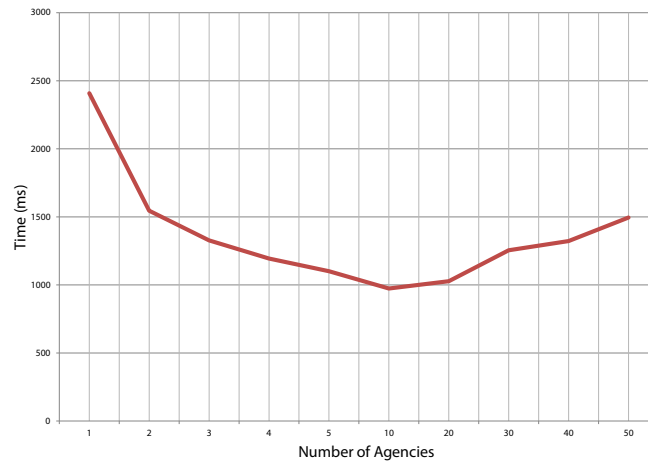


Figure 4. Number of agents \times Response time for 100000 registered taxis.

In Figure 4, it can be noted that the response time decreases when we increase the number of agencies treating the same client request up to 10 agents, after 10 the response time increases slightly for 20, 30, 40, and 50 agencies. The results show that the communication costs no longer compensate given the little time needed by agents to find the best local taxi, when the number of agencies agents is too large.

It is important to mention that the experiments were executed in a local computer and we have not yet used the mobile network to evaluate the geometric treatment for client requests. However, we believe that the overall results will be similar when we experiment with actual mobile phones. We aim to do so in future work.

5. Related Work

In [18], the authors propose a multi-agent self-organisation approach for the task of dispatching taxis to clients, in order to improve the performance of the current organisation of a real taxi company. The company operates dividing the city into regions, and the calls that are made by clients inside some of these regions will be preferably dispatched for an available taxi that is also currently within that region. Such approach aims to minimise both the waiting time for the client and the driver time to reach the client location. In the case when there are no available taxis in the client region, the system searches for available taxis on adjacent areas, where these adjacency relations were required to be manually specified in the system. The multi-agent self-organisation approach implemented aims to dynamically establish the adjacency relations between the regions in a way to make better use of the information about the current distribution of the taxis in the areas and changes in the traffic patterns. Our work differs from [18] in that we aim to demonstrate the various features of multi-agent systems that could be applied in such scenarios. We treat the

client requests differently, as the geometric treatment we propose allows multiples agents to treat a client request, finding the best global taxi available in the system. Distributing the treatment for client requests, our results show a reduction of 50% on the time needed for agents to find the best global solution. Further, our work differs in the evaluation, and we argue that our approach could have in impact on the results presented in [18], improving the response time of that work.

In [1], the authors propose a multi-agent system in which the agents, in that work associated with the taxis, are capable of negotiating their requests after they have been assigned to them, allowing the taxis to exchange customers that have not been collected yet, maximising the average satisfaction within groups of requests. The system operates grouping the client requests that happened within a time window and assigning each of these groups to a number of taxis that are available at that time. Then the system applies a decentralised algorithm which works dividing its reasoning process into a number of rounds of negotiation among the taxi agents. In those rounds, the taxi agents are able to exchange their assigned requests in order to improve the average sum of quality-of-service among all the assignments in the application. Our work differs from [1] as we are interested in reducing the response time of the system as much as we can, differently from [1] which uses time windows to process and respond requests. Also, as described before, our evaluation concerns are specifically in the system responding timely, which could improve the approach in [1], including scalability issues.

In [2], the authors propose a practical implementation of a multi-agent system for the dispatching of taxis to costumers in a big taxi company that operates in London. Up until then, the company used to manually handle the incoming client requests and the management of the fleet. Because this was a real-world case, the authors had to deal with a number of complicated events regarding scalability, unpredictable events, and variety in customer needs *versus* variety of vehicles. The system described in [2] is composed of a number of modules interconnected by event queues, and makes use of a scheduling approach named “adaptive scheduling”. The key feature of this technique is that when any disruptive event occurs the system does not need to rebuild all the schedule from scratch, but only specific parts that may have been affected by the event. The system is capable of scheduling the mapping of taxis to customers based on a series of criteria, such as the physical distance to the costumer, traffic, type of client, as well as handling situations like a possible order cancellation and idle time of the drivers, being able to improve both client and driver satisfaction. Our approach uses a more dynamic treatment for taxis available than [2], and we deal with scalability and self-organisation, which is not treated in that work. Our approach could benefit from the other criteria, such as traffic, type of client, etc., used in [2] for the allocation of taxis to clients. We used only the distance in this first prototype, because our goal here was to describe the overall architecture and features of our innovative approach. We argue that [2] could also benefit from our approach in aspects such as distribution, scalability, geometric treatment, etc.

In [4], the authors explain the need for dynamic and distributed methods for solving problems that handle dynamic behaviour, such as courier services and taxi services. The main reasons that there are few deployed multi-agent systems in today’s industry are mostly because multi-agent systems are usually not optimised for a specific problem and they cannot guarantee to perform well. Engineering such a system is hard because

all of its layers can mutate in a non-linear way, given that there is no fully established method for engineering specific multi-agent systems that can control or give guarantees about its emergent properties. Furthermore, it is believed that to further the development of research in dynamic logistic problems, a set of standards for benchmarking multi-agent systems need to be consolidated, so that the scientific quality of papers in this field can be improved. We take some inspiration from [4] in this work, including the evaluation we propose. As described in [4], it is difficult to find datasets and benchmarks, so we have generated our own datasets to evaluate our application through simulation. Also, it is difficult to compare our approach with others, given that they have focused on another kind of evaluation.

All papers discussed in this section relate to the scenario we have investigated and the architecture we propose, but we argue that different scenarios could take advantage from our approach. Also, we argue that the overall approach could include other techniques to improve the application, such as richer communication models (e.g. argumentation-based techniques) in order to solve the problem of reallocation, fault tolerance techniques, and so forth.

6. Final Remarks

In this work, we described a taxi service application built on the multi-agent paradigm. The system is able to self-organise in the sense of the geometric treatment we proposed and in regards to scalability issues. We argue that multi-agent systems fit well in the problem described in this work, where we have a distributed problem with highly dynamic and unpredictable environment and entities. Also, such kind of application suffers with network and server related problems, which could be appropriately handled by multi-agent systems, as described in this work. The contributions in this work were: (i) we proposed a taxi service application, exploring the main advantages of using multi-agent systems; (ii) we evaluate our application in regards to the response time for different configurations regarding the treatment of a client request by multiple agencies. Our results show that the scalability achieved in this work is necessary, as well as it improves the system response time. However, increasing the number of agents to handle scalability should be done with care as communication can start to compromise performance in a subtle way, and the system could check the communication overheads in the response time and self-organise based on such checks.

References

- [1] K. T. Seow, N. H. Dang, and D.-H. Lee, “A collaborative multiagent taxi-dispatch system,” *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 607–616, 2010.
- [2] A. Glaschenko, A. Ivaschenko, G. Rzevski, and P. Skobelev, “Multi-agent real time scheduling system for taxi companies,” in *AAMAS*, 2009, pp. 29–36.
- [3] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [4] R. R. van Lon and T. Holvoet, “Evolved multi-agent systems and thorough evaluation are necessary for scalable logistics (position paper),” in *Computational Intelligence In Production And Logistics Systems (CIPLS)*, pp. 48–53, 2013.

- [5] R. H. Bordini, M. Dastani, J. Dix, and A. E. F. Seghrouchni, *Multi-Agent Programming: Languages, Tools and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2009.
- [6] R. H. Bordini, J. F. Hübner, and M. Wooldridge, *Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons, 2007.
- [7] A. S. Rao, “AgentSpeak(L): BDI agents speak out in a logical computable language,” in *Proceedings of the 7th European workshop on Modelling autonomous agents in a multi-agent world : agents breaking away: agents breaking away*, ser. MAAMAW '96. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996, pp. 42–55.
- [8] A. R. Panisson, F. Meneguzzi, M. Fagundes, R. Vieira, and R. H. Bordini, “Formal semantics of speech acts for argumentative dialogues,” in *AAMAS*, 2014, pp. 1437–1438.
- [9] A. R. Panisson, F. Meneguzzi, R. Vieira, and R. H. Bordini, “Towards practical argumentation in multi-agent systems,” in *Brazilian Conference on Intelligent Systems, BRACIS*, 2015.
- [10] A. Ricci, M. Piunti, and M. Viroli, “Environment programming in multi-agent systems: An artifact-based perspective,” *AAMAS*, vol. 23, no. 2, pp. 158–192, Sep. 2011.
- [11] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, and A. Santi, “Multi-agent oriented programming with jacamo,” *Science of Computer Programming*, vol. 78, no. 6, pp. 747–761, 2013.
- [12] J. F. Hübner, O. Boissier, and J. S. Sichman, “Programming mas reorganisation with moise+,” in *Dagstuhl Seminar on Foundations and Practice of Programming Multi-Agent Systems*, vol. 6261, 2006.
- [13] A. Santi, M. Guidi, and A. Ricci, “Jaca-android: an agent-based platform for building smart mobile applications,” in *International Workshop on Languages, Methodologies and Development Tools for Multi-Agent Systems*. 2010, pp. 95–114.
- [14] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, and A. Santi, “Multi-agent oriented programming with jacamo,” *Science of Computer Programming*, vol. 78, no. 6, pp. 747–761, 2013.
- [15] A. Sorici, O. Boissier, G. Picard, and A. Santi, “Exploiting the jacamo framework for realising an adaptive room governance application,” in *SPLASH '11 Workshops*, ACM, 2011, pp. 239–242.
- [16] M. S. Fagundes, F. Meneguzzi, R. Vieira, and R. H. Bordini, “Interaction patterns in a multi-agent organisation to support shared tasks,” in *International and Interdisciplinary Conference on Modeling and Using Context*. Springer Berlin Heidelberg, 2013, pp. 364–370.
- [17] D. Schmidt, A. R. Panisson, A. Freitas, R. H. Bordini, F. Meneguzzi, and R. Vieira, “An ontology-based mobile application for task managing in collaborative groups,” in *FLAIRS*, 2016.
- [18] A. Alshamsi, S. Abdallah, and I. Rahwan, “Multiagent self-organization for a taxi dispatch system,” in *AAMAS*, 2009, pp. 21–28.

Um Algoritmo para Agrupamento de Dados Usando Interações entre Agentes

Lutiele M. Godois¹, Luciano C. Marco¹, Diana F. Adamatti¹, Leonardo R. Emmendorfer¹

¹ Programa de Pós-Graduação em Modelagem Computacional - Centro de Ciências Computacionais - C3 – Universidade Federal do Rio Grande (FURG)
Caixa Postal 474 – 96.201-900 – Rio Grande – RS – Brazil

lutielegodois@furg.br, lucianomarco@msn.com,
leonardo.emmendorfer@gmail.com, dianaada@gmail.com

Abstract. *The Clustering task in data mining consists of grouping data objects that are similar to each other. Many techniques for this purpose have been proposed, most of them based on a priori information to get the results, such as the desired number of clusters. Thus, this article presents a clustering algorithm based on a multi-agent system, which automatically detects clusters for a given set of data, which are emerging patterns of agent interaction. The results show the convergence of the algorithm for some data sets, which illustrate some classic clustering problems.*

Resumo. *A tarefa de Agrupamento em Mineração de Dados consiste em agrupar objetos de dados que são semelhantes entre si. Muitas técnicas para esse fim foram propostas, a maioria delas se baseia em informações a priori para obter os resultados, como o número desejado de clusters. Assim, este artigo apresenta um algoritmo de agrupamento baseado em um sistema multiagente, que detecta automaticamente os grupos para um determinado conjunto de dados, que são padrões emergentes da interação entre agentes. Os resultados mostram a convergência do algoritmo para alguns conjuntos de dados, que ilustram alguns problemas clássicos de agrupamento.*

1. Introdução

Em mineração de dados, a tarefa de agrupamento (Clustering) tornou-se um tema cada vez mais importante nos últimos anos, devido ao grande número de dados a ser manipulados nas diversas áreas: biologia, medicina, psicologia, processamento de imagens, etc.

Os Algoritmos de agrupamento têm como objetivo particionar os objetos de dados (padrões, entidades, instâncias, observações, unidades) em certo número de clusters (grupos, subconjuntos ou categorias) [Xu e Wunsch II 2009]. Devido ao grande número de aplicações, dificilmente se encontrará uma técnica de clustering genérica e aplicável de forma satisfatória a todos os tipos de dados nos diferentes contextos, logo, existem muitos algoritmos. São alguns exemplos de algoritmos para agrupamento: o K-means [MacQueen 1967], o DBSCAN [Ester et al. 1996] e o CURE [Guha, Rastogi and Shim 1998]. Dessa forma, apresenta-se nesse trabalho alguns resultados de uma nova proposta de algoritmo baseado em um Sistema Multiagente.

Os Sistemas Multiagente são compostos por múltiplos elementos computacionais que se interagem, conhecidos como agentes, eles por sua vez, possuem a capacidade de ação autônoma, de decidir por si mesmo o que precisam fazer para satisfazer seus objetivos, além poderem interagir com outros agentes [Wooldridge 2002]. Assim, é justamente essa possibilidade de autonomia, interação, cooperação e troca de informações entre agentes que a técnica de clustering apresentada aqui se baseia, com o intuito de desenvolver um algoritmo capaz de realizar a auto-organização de diferentes conjuntos de objetos de dados. Nesse algoritmo, cada agente (objeto de dado) realiza certa tarefa com alguns critérios definidos para formar grupos com outros agentes, baseado apenas em sua localização espacial e sua capacidade de “enxergar” seus vizinhos. O critério principal utilizado é que o agente escolha o grupo com o maior número de agentes para se agrupar de acordo com um alcance de visão definido para ele. Para cumprir esse objetivo, a implementação do algoritmo e os testes foram realizados na plataforma NetLogo.

Nesse primeiro momento, o algoritmo é aplicado em diferentes conjuntos de dados encontrados em Ultsch (2005), com grupos de formatos diferentes, pois ainda não se podem afirmar quais tipos de problemas de agrupamento de dados a nova técnica melhor se aplica. O problema central, do trabalho, dessa forma é verificar se o Sistema Multiagente desenvolvido pode ser considerado uma boa ferramenta para agrupamento de dados, através da aplicação do algoritmo nesses conjuntos de dados que correspondem a problemas que podem ser encontrados dados reais.

Esse artigo está estruturado da seguinte maneira: na Seção 2 apresentam-se alguns trabalhos relacionados com o tema do trabalho; na Seção 3 são apresentados o algoritmo de agrupamento proposto e a metodologia utilizada no processo de implementação; a Seção 4 apresenta os resultados alcançados; e por fim na Seção 5 estão as conclusões do trabalho.

2. Trabalhos Relacionados

Agrupamento de dados é um problemas amplamente estudado, o qual possui numerosas aplicações. O problema básico de agrupamento de dados pode ser afirmado da seguinte forma: dado um conjunto de pontos de dados, divida-os em um conjunto de grupos que são tão semelhantes quanto possível. Porém essa definição de problema pode sofrer algumas alterações dependendo do modelo específico utilizado e também o tipo de dado. Por exemplo, um modelo generativo pode definir similaridade com base em um mecanismo generativo probabilístico, enquanto que uma abordagem baseada em distância utilizará uma função de distância tradicional para quantificação [Aggarwal e Reddy 2014].

Na literatura são encontrados muitos métodos para mostrar a similaridade e dissimilaridade entre grupos de dados. Talvez o mais tradicional de todos seja o K-means. Esse método consiste em particionar um conjunto de objetos de dados em k grupos, definido a priori, além disso, é definido um centroide para cada cluster. É a partir desse centroide que se buscam os objetos mais próximos a ele, obtendo-se assim os clusters. A atualização desses centros é realizada sempre que um novo objeto é atribuído ao cluster, sendo o critério de parada desse algoritmo quando não há mais alterações. As escolhas iniciais de número de grupos e centroides se constituem com uma das desvantagens desse algoritmo, além de outros que possuem uma perspectiva

centralizada, pois podem afetar a qualidade agrupamento final, não identificando alguns grupos naturais nos conjuntos de dados, de forma que esse processo é condicionado a esses critérios.

Contrário a essas propostas centralizadas de informar muitos dados de entradas e principalmente a necessidade de especificar o número de grupos desejados, se propõe neste trabalho um algoritmo de agrupamento que possui como principal característica a auto-organização de objetos em grupos segundo o critério determinado para este fim. Além disso, o algoritmo também se embasa na concepção de Sistema Multiagente, onde cada objeto de dado é compreendido com um agente que deve realizar uma tarefa para se organizar em grupos com os outros agentes que pertencem ao conjunto.

Algoritmos de Agrupamentos baseados em Sistema Multiagente já são temas de alguns estudos. Santos e Bazan (2012) propõe em seu trabalho um algoritmo de agrupamento inspirado em técnicas de inteligência de enxames, como a organização de colônias de abelhas e a alocação de tarefas entre insetos sociais, partindo também do pressuposto contrário das técnicas tradicionais que necessitam de algumas orientações para a alocação de dados em grupos. O processo de tomada de decisão dos agentes é inspirado no trabalho de Agogino e Tumer (2006). Esses autores, por sua vez, apresentam uma alternativa quando se deseja trabalhar com numerosos domínios. Nesta abordagem, para cada agente é atribuído um pequeno subconjunto dos dados e esses dados escolhem em qual grupo final seus pontos de dados devem pertencer. O agrupamento final é então avaliado por uma função utilitária global, calculada de forma distribuída. Os resultados obtidos nesse trabalho mostram que essa metodologia específica para agente pode obter um melhor desempenho do que os métodos tradicionais não baseados em agentes.

Já Takacs e Demiris (2010) descrevem um novo algoritmo que utiliza subdivisões iterativas onde os clusters são pré-processados em cada etapa para combater o dimensionamento espacial, a rotação, a velocidade de repetição e a frequência de amostragem variável. Além disso, apresentam um método para equilibrar a segmentação espacial e temporal com base no tamanho de grupo esperado, e uma medida de validade é introduzida para determinar o número ótimo de clusters. A aplicação foi realizada em jogos e os resultados comparados com o K-means e o tradicional Clustering Espectral.

Chaimontree, Atkinson e Coenen (2012) descrevem um framework para agrupamento baseado em multiagentes onde cada agente individual representa um cluster individual. Uma característica particular do framework é que, após a criação de uma configuração de cluster inicial, os agentes podem negociar com o objetivo de melhorar esse cluster inicial. O framework pode ser usado no contexto de um número de paradigmas de agrupamento, dois são investigados: K-means e KNN. A avaliação relatada demonstra que a negociação pode servir para melhorar uma configuração de cluster inicial.

Gueleri (2013) desenvolveu técnicas de agrupamento baseadas em comportamento coletivo e auto-organização. Nessas técnicas cada objeto do conjunto de dados corresponde a um indivíduo do sistema, os quais interagem um com os outros, de forma que os grupos apareçam a partir de uma organização promovida por próprios. Os objetos são mantidos fixos em seu espaço de atributos, mas carregam certo tipo de "energia". Tal energia será trocada gradualmente entre eles. Logo os grupos serão formados por objetos que possuem energias semelhantes.

Os exemplos anteriores exemplificam alguns métodos para agrupar objetos de dados dentro de uma concepção de agente e auto-organização. A proposta apresentada neste artigo pretende salientar ainda mais esses conceitos, dando ênfase nas propriedades que os agentes possuem, como a autonomia e a capacidade de reação. Porém, ao contrário de algumas propostas expostas anteriormente a técnica a ser desenvolvida não se baseará em algum comportamento bio-inspirado, as características para cada agente (objeto de dado) serão formuladas de maneira artificial e os grupos serão formados a partir da capacidade visual de cada agente e de suas movimentações durante o processo de agrupamento. Nas próximas seções são expostas as etapas e como ocorre a interação do agente no processo de agrupamento.

Para fins de avaliação do algoritmo proposto, não é considerado nesse primeiro momento as técnicas abordadas nos trabalhos correlatos. Dessa forma, ele será avaliado a partir dos resultados encontrados no trabalho que descrevem os conjuntos de dados utilizados.

3. Metodologia

Nessa seção é detalhado o algoritmo, conforme apresentado no pseudocódigo representado na Figura 1. Para realizar uma análise experimental, programou-se uma versão do algoritmo utilizando o software NetLogo. Esse ambiente de programação é adequado para a modelagem de sistemas complexos e possuem um desenvolvimento ao longo do tempo.

Entrada: Um conjunto de dados (agentes) X

Saída: Uma partição X em k grupos

```

início
   $r = \max d(x_i, x_j)$  // maior distância entre dois agentes
  para cada  $x_i \in X$  faça
     $c_i = i$ 
    // cada agente está no seu próprio grupo.
  fim
  repita
    repita
      para cada  $x_i \in X$  faça
        De acordo com a visão dada encontrar a direção que possui mais agentes;
        if Houver mais de uma direção com o mesmo número de agentes
          then
            Mover agente aleatoriamente;
          else
            Mover o agente na direção com mais agentes;
             $c_i = c_a$ 
            //  $x_i$  herda o rótulo de qualquer  $x_a$  pertencente ao grupo que é direcionado.
        end
      fim
    até Não haver mais alteração nos agrupamentos;
    Retornar os  $x_i$  a sua posição inicial com o novo rótulo.
     $r = r - 1$ 
  até Até que o número final de grupos seja maior que 1;
fim

```

Figura 1. Pseudocódigo do Algoritmo Proposto

O algoritmo proposto é iniciado com a entrada de um conjunto de objetos de dados, definindo um objeto de dado como um agente. A variável r permite buscar o valor do alcance de visão dos agentes, de modo que este valor seja o mais adequado para cada conjunto de dados. Inicialmente, com $r = \max d(x_i, x_j)$, a visão dos agentes é máxima e todos “enxergam-se”, levando imediatamente a uma solução trivial, com um único grupo. O valor de r é reduzido gradativamente até que uma solução não trivial, com mais do que apenas um grupo, seja encontrada.

Nesta implementação, o campo de direção de 360° foi discretizado para apenas 8 setores, cada um cobrindo uma faixa de 45° . Em cada iteração, a direção para onde os agentes se movem é definida usando o critério baseado no número de outros agentes percebidos por cada agente x_i , sendo que eles se movem em uma velocidade constante. Se não houver uma melhor direção percebida para se mover (ou seja, sem agentes em qualquer direção ou em caso de empates), uma direção aleatória é definida. Caso contrário, a direção de x_i é definida para onde o número de vizinhos é maior.

O rótulo de cluster de cada agente, que foram considerados diferentes inicialmente, será alterado quando x_i mover-se para a direção com mais agentes. Assim, x_i herda o rótulo de qualquer x_a pertencente ao cluster que é direcionado. Nesse caso, ambos os agentes estão configurados para o mesmo cluster, pois o rótulo de x_i está configurado para ser o mesmo rótulo de x_a .

Dessa forma, esse algoritmo se destaca dos demais quando não se faz necessário informar o número de grupos antes da execução do algoritmo. O número de grupos será formado automaticamente durante o processo de agrupamento, de acordo com os alcances de visão dos agentes.

Esse algoritmo foi aplicado nos seguintes conjuntos de dados de duas dimensões encontrados em Ultsch (2005), denominados FCPS (*Fundamental Clustering Problem Suite* - Conjunto de Problemas Fundamentais de Agrupamentos de Dados): Lsun, EngyTime, Target, TwoDiamonds e WingNut que possuem respectivamente os seguintes tamanhos, ou número de pontos: 400, 4096, 770, 800 e 1070. Os problemas abordados em cada conjunto são explicados na seção dos resultados juntamente com os agrupamentos finais gerados após a aplicação do algoritmo proposto. Porém, outros resultados preliminares do desempenho do algoritmo podem ser vistos em Godois et al. (2017), considerando apenas conjuntos de dados gerados de forma aleatória.

Inicialmente, com o intuito apenas comparativo, os agrupamentos gerados são validados utilizando o índice interno de validação *Silhouette* [Rousseeuw 1987], mostrado na Equação (1).

$$s(v_i) = \frac{d(v_i, C_h) - d(v_i, C_j)}{\max(d(v_i, C_h), d(v_i, C_j))} \quad (1)$$

Onde, $d(v_i, C_h)$ é a dissimilaridade média do objeto v_i em relação a todos os outros objetos do grupo mais próximo C_h ; $d(v_i, C_j)$ é a dissimilaridade média do objeto v_i em relação a todos os outros objetos do grupo C_j , que contém v_i e $s(v_i)$ varia no intervalo fechado $[-1, 1]$, de tal forma que quando mais próximo de 1, melhor a alocação

objeto no grupo. Após este cálculo para os objetos de dados, é necessário calcular a média para o agrupamento como um todo.

As médias para o índice encontrado para cada agrupamento obtido aos conjuntos de dados são comparadas com as médias encontradas para os agrupamentos finais realizados pelo algoritmo U*C, apresentados no trabalho de Ultsch (2005). Em seu trabalho, o autor considera apenas uma classificação como correta. Aqui, tenta-se encontrar outras soluções que possa ser considerada “satisfatória” com a nova proposta algoritmo baseada na interação de agentes.

4. Resultados

Para cada conjunto de dados, os testes foram realizados considerando diferentes alcances de visão diferente dos agentes. A Tabela 1 e 2 mostra, alguns resultados para cada conjunto de dados após sua aplicação em nosso algoritmo em comparação do algoritmo U*C. Nessa tabela, apenas se encontram os melhores resultados considerando a maximização do índice *Silhouette* ($Max. s(v_i)$) calculado. É importante notar que, para alguns conjuntos de dados, os testes não foram realizados para a distância da visão até 9, pois o algoritmo não se aplica quando um único grupo é obtido para o conjunto.

Tabela 1. Comparação dos resultados da aplicação do Algoritmo U*C nos conjuntos de dados com o melhor resultado obtido com o algoritmo proposto conspirando a maximização do índice

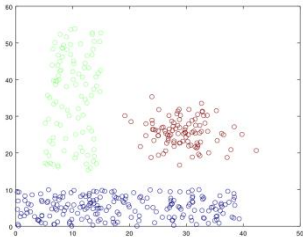
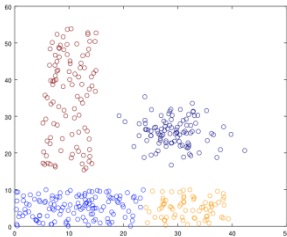
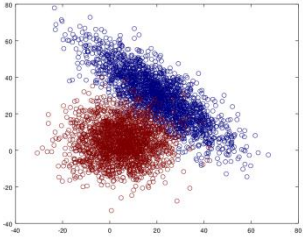
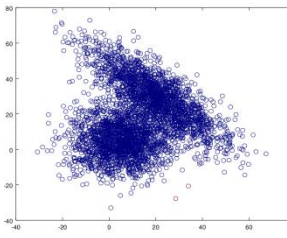
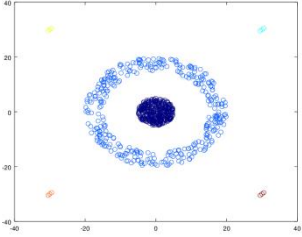
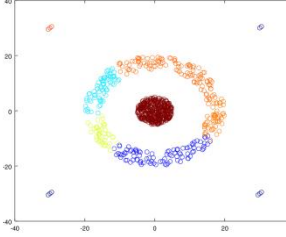
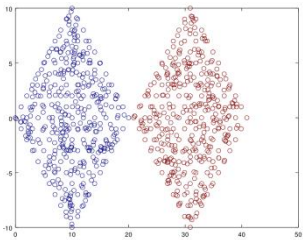
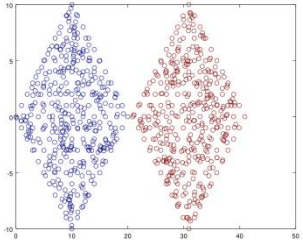
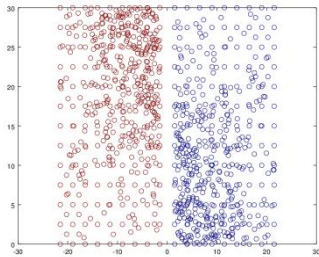
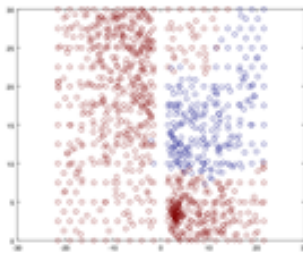
Conjunto de Dados	Algoritmo U*C	Algoritmo proposto
Lsun		
EngyTime		
Target		

Tabela 2. Comparação dos resultados da aplicação do Algoritmo U*C nos conjuntos de dados com o melhor resultado obtido com o algoritmo proposto conspirando a maximização do índice

Conjunto de Dados	Algoritmo U*C	Algoritmo proposto
TwoDiamonds		
WingNut		

O conjunto de dados Lsun tem como problema principal as diferentes variâncias e as distâncias inter-cluster. Na Tabela 3, se tem os resultados encontrados para a classificação dada por Ultsch (2005) e na Tabela 4, são encontrados os resultados considerando o algoritmo proposto neste trabalho. Comparando os resultados, com o novo algoritmo, obtemos dois (alcance de visão do agente 4 e 5), agrupamentos melhores e dois resultados semelhantes ao U*C.

O EngyTime tem como problema uma mistura gaussiana. Comparando os resultados, o algoritmo U*C obteve uma média $s(v_i)$ melhor que o algoritmo propostos. Com a nova técnica os melhores resultados são obtidos para a distância 7 e 9 da visão do agente.

O conjunto de dados Target tem como principal problema os *outliers*, foi o que teve melhor desempenho com o algoritmo de interação de agentes. Na comparação dos resultados, nova proposta de algoritmo obteve apenas um resultado menor do que o outro algoritmo, para o alcance de visão 1.

Para o conjunto TwoDiamonds cujo problema principal são as bordas dos cluster definido pela densidade, tem-se que os dois algoritmos obtiveram resultados semelhantes quando considerados os alcances de visão do agente 3, 4 e 5 no algoritmo proposto.

Por fim, o conjunto de dados WingNut que apresenta problemas relacionados com a densidade e a distância. O algoritmo U * C obteve uma média $s(v_i)$ melhor, em comparação com os resultados do algoritmo apresentado nesse trabalho.

Tabela 3. Resultados para o cálculo do índice interno *Silhouette* index para os conjuntos de dados considerando o algoritmo U*C (Ultsch, (2005)).

Conjunto de dados	Média dos $s(vi)$
Lsun	0.60921
EngyTime	0.55642
Target	0.26034
TwoDiamonds	0.83180
WingNut	0.63031

Tabela 4. Resultados para o cálculo do índice interno *Silhouette* index para os conjuntos de dados considerando o algoritmo proposta baseado na interação de agentes.

Alcance de visão dos agentes	Média dos $s(vi)$	Média dos $s(vi)$	Média dos $s(vi)$	Média dos $s(vi)$	Média dos $s(vi)$
	<i>Lsun</i>	<i>EngyTime</i>	<i>Target</i>	<i>TwoDiamonds</i>	<i>WingNut</i>
1	0.39172	-0.36197	-0.19056	-0.45463	-0.36403
2	-0.25728	-0.11184	0.49609	0.54030	-0.046701
3	0.32095	-0.05660	0.69037	0.83192	0.14579
4	0.64119	0.03981	0.68265	0.83192	0.15979
5	0.68492	-0.03785	0.47348	0.83224	-
6	0.60909	0.01096	0.51777	-	-
7	0.60897	0.44589	0.42590	-	-
8	0.31756	0.31233	0.26033	-	-
9	0.20478	0.47886	0.26156	-	-
Max. $s(vi)$	0.68492	0.47886	0.69037	0.83224	0.15979
Melhor alcance de visão	5	9	3	5	4

5. Conclusões

Com base nos resultados obtidos, acreditamos que nosso algoritmo baseado em agente pode ser usado para vários problemas. Desta forma, não se limita a um único problema ou solução específica. Além disso, o usuário não precisa definir centroides, como alguns algoritmos tradicionais, para definir grupos, pois eles se organizam, formando seus próprios grupos sem informações iniciais sobre o número desejado. Além disso, o algoritmo segue algumas propriedades do agente, como a autonomia que contribui para a tomada de decisão durante o processo de agrupamento. No entanto, ele é dependente de determinados parâmetros, como o alcance de visão do agente.

Neste artigo, obtemos o melhor valor do alcance de visão do agente para cada conjunto de dados, maximizando o índice *Silhouette*, mas outras maneiras de encontrar o valor ideal podem ser testadas. Visto que, em muitas aplicações práticas, os índices mais tradicionais, como o *Silhouette*, não são capazes de quantificar corretamente as soluções para agrupamentos de dados, ou seja, muitas dessas medidas descrevem melhores resultados para grupos de forma convexa e não são bons suficientes para validação de grupos não convexos.

Novos estudos, análises e propostas para melhorar seu desempenho e estabilidade serão realizados, de forma que se torne uma boa ferramenta para a mineração de dados, como por exemplo: considerar a densidade de pontos em certo local para encontrar o melhor alcance de visão para os agentes, ou ainda considerar outros valores reais positivos diferentes dos inteiros e positivos utilizados nesse algoritmo, com o objetivo de obter melhores geometrias para cada grupo formado.

Pelos testes no NetLogo, o algoritmo obteve um comportamento rápido, em trabalhos futuros, haverá mais ênfase a esse aspecto. Além disso, outros conjuntos de dados e índices de validação de cluster serão utilizados, bem como a comparação com outras técnicas de clustering.

6. Agradecimentos

Os autores agradecem especialmente a Capes (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro, colaborando assim para a realização da pesquisa.

Referências

- Aggarwal, C. C. and Reddy, C. K. (2014) “Data Clustering: Algorithms and Applications”. CRC Press, NY, USA.
- Agogino, A. and Tumer, K. (2006) “Efficient agent-based cluster ensembles”. In Proc. 5th Int. Conf. on Autonomous agents and multiagent systems. AAMAS '06, ACM, New York, , p. 1079–1086.
- Chaimontree, S.; Atkinson, K. and Coenen F. (2012) “A Multi-Agent Based Approach To Clustering: Harnessing The Power of Agents”. In Agents and Data Mining Interaction, Vol. 7103 of the series Lecture Notes in Computer Science, p. 16-29.

- Ester, M.; Kriegel, H.-P.; Sander, J.; XU, X. (1996) “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.” In KDD-96 Proceedings, p.226–231.
- Godois, L. M.; Marco, L. C.; Adamatti, D. F.; Emmendorfer, L. R. (2017) “Algoritmo DL3: Uma abordagem de Clustering baseado em Auto-organização.” In Proceedings of WESAAC 2017, p. 90-100.
- Gueleri, R. A.; (2013) “Agrupamento de dados baseado em comportamento coletivo e auto-organização”. Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional - Mestrado em Ciências da Computação e Matemática Computacional — Universidade de São Paulo, São Carlos.
- Guha, S.; Rastogi, R.; Shim, K. (1998) “CURE: An Efficient Clustering Algorithm for Large Databases”. In SIGMOD, Vol.27, p.73–84.
- MacQueen, J. (1967). “Some methods for classification and analysis of multivariate observations”. In *Proceedings of the Fifth Berkeley Symposium*, Vol. 1, p. 281-297.
- Rousseeuw, P. J. (1987) “Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis”. In *Computational and Applied Mathematics*, Vol.20, p.53–65.
- Santos, D. S. and Bazzan, A. L. C. (2012) “Distributed clustering for group formation and task allocation in multiagent systems: A swarm intelligence approach”. In *Appl. Soft Comput.*, Vol. 8, p. 2123-2131.
- Takacs, B. and Demiris, Y. (2010) “Spectral clustering in multi-agent systems”. In *Knowl Inf. Syst.*, Vol. 25, p. 607–622.
- Ultsch, A. (2005) “Clustering with SOM: U*C”. In *Proceedings 5th Workshop on Self-Organizing Maps – WSOM 2005, Paris*, p.75–82.
- Wooldridge, M. (2002) “An Introduction to Multiagent Systems”, JOHN WILEY & SONS, LTD, 1st edition.
- Xu, R. and Wunsch II, D. C. (2009) “Clustering”, IEEE Press, 1st edition.

Analysis of unsupervised aspect term identification methods for Portuguese reviews

Mateus Tarcinalli Machado^{1,2,4}, Thiago Alexandre Salgueiro Pardo^{2,4},
Evandro Eduardo Seron Ruiz^{3,4}

¹Programa de Pós-Graduação em Computação Aplicada

²Departamento de Ciências da Computação, ICMC

³Departamento de Computação e Matemática, FFCLRP

⁴Universidade de São Paulo, Brazil.

[mateusmachado, taspardo, evandro]@usp.br

Abstract. *Automatic term identification is a fundamental task for text mining and many other natural language processing applications. Based on the tradition of statistical and symbolic methods, whether adopting or not machine learning techniques, new hybrid proposals for term extraction have arisen. In this paper we investigate the above approaches in an unsupervised way to identify relevant terms in Portuguese, focusing on book reviews. Terms, in this context, usually represent opinion aspects, whose identification is a challenging and an equally important task. We show that current methods have varied performances, but that we may significantly improve their results by considering the distributional properties of words.*

Resumo. *A identificação automática de termos é fundamental na mineração de textos e em aplicações de processamento de linguagem natural. Com base em métodos estatísticos e simbólicos, adotando ou não técnicas de aprendizado de máquina, surgiram novas propostas híbridas para extração de termos. Neste trabalho, investigamos essas abordagens de forma não supervisionada para identificar termos relevantes em Português, com foco em revisões de livros. Os termos, neste contexto, geralmente representam aspectos de opinião, cuja identificação é uma tarefa desafiadora e igualmente importante. Mostramos que métodos atuais têm desempenhos variados, mas que seus resultados podem ser melhorados considerando as propriedades distribucionais das palavras.*

1. Introduction

Besides programming machines with basic linguistic knowledge and communication skills, one of the main obstacles for a comprehensive human-computer interaction is the lack of ability to fully understand human language in a particular context. This is true even for human communication, as many different daily situations (from simpler ones, as going to a social meeting or sharing a comment about a product on the web; to more complex ones, as negotiating the details of a business) demand specific knowledge about the circumstances and the usual communication protocol to follow. For computers to perform well in such conditions, they need to acquire the necessary domain knowledge, as humans do.

A small but significant step towards this aspiration may be given by the field of Automatic Term Extraction, also known by the ATE acronym. Term extraction is the task of identifying a vocabulary, or relevant words in a domain [Frakes et al. 2014]. Recently, ATE has gained special attention in several tasks in the Natural Language Processing (NLP), Machine Learning and related areas, including, e.g., applications in biomedical informatics, where ATE is used for translation of technical terms [Lossio-Ventura et al. 2016], and applications that use ATE on electronic textbooks for creating educational ontologies for learning systems [Conde et al. 2016]. In particular, ATE is very relevant in the more recent area of opinion mining and sentiment analysis, more specifically, in the area of aspect-based sentiment analysis, which aims at analyzing a large amount of data and synthesizing relevant information about aspects evaluated by users (e.g., the technical aspects of smartphones, as screen quality and battery life). In this context, terms usually represent aspects, and the precise identification of aspects of the evaluated entity is crucial to accurately assign them their corresponding sentiments/polarities [Schouten and Frasinca 2016, Pavlopoulos and Androutsopoulos 2014].

The importance of opinion mining and sentiment analysis has increased with the Web 2.0 and the upload of user generated text in various on-line services, such as chat/messengers, blogs and micro-blogs, social media communication, e-commerce and product review sites. All these services contributed to such an amount of unstructured data that demanded new ways to extract, load, transform and analyze the information content [Ravi and Ravi 2015]. Nowadays, understanding the opinions of the public on distinctive matters, such as political engagement, product preferences and social events, to name only a few, is an industry concern and also an academic research opportunity. It is also a relevant task to the final customer in situations where s/he needs supporting information for taking a decision about buying a product, reading a book, or subscribing to a service, among other situations. Correctly identifying the *aspect terms* is very important, therefore.

This paper investigates three different methods for ATE, as well as some variations and combinations. We investigate: 1) a frequency-based baseline method (which we refer by *FREQ*); 2) the Hu & Liu hybrid method [Hu and Liu 2004b]; and 3) the enhancement of the two previous methods with a word distributional approach, the *word2vec* method [Mikolov et al. 2013] (referred by *W2V* from here on). We evaluate how good the methods are in finding the aspects in book reviews written in Portuguese. This paper presents the following novelties related to the state-of-the-art in ATE: a) it evaluates the three above ATE approaches in the first and only known Portuguese annotated corpus for aspects; and b) the Hu & Liu method was enhanced by processing infrequent terms with the *word2vec* approach. We show that the three researched methods for ATE present varied performances, but that the distributional properties of words may improve the results.

Section 2 reports some related research efforts. In Section 3, the data and the ATE methods that we apply are explained. Experimental results are reported and discussed in Section 4. Some conclusions and final remarks are presented in Section 5.

2. Related work

One of the first comprehensive researches about ATE dates back to middle 1990, when Kageura and colleagues [Kageura and Umino 1996] report about a number of approaches to what they called automatic term recognition (ATR). At the time, the techniques adopted word statistics to generate a list of index terms for a given document. Statistical approaches explore the fact that the words composing a term tend to be found closer to each other. Later, Castellvi and collaborators [Castellví et al. 2001] analyzed twelve terminology extraction tools, including symbolic and hybrid ATE approaches.

Frantzi and Ananiadou [Frantzi and Ananiadou 1997] suggest a C-value method, using statistical weights, for the extraction of multi-word terms. We know the majority of domain specific terms consist of multiple words, also known as uninterrupted collocations. The C-value method incorporates context information for the evaluation of candidate terms. It was reported later [Frantzi et al. 2000] that the C-value method increased the precision on selected terms from figures of 30% to figures in a range of 69–74%. About 85% of domain specific terms are said to be compound nouns [Nakagawa and Mori 2002].

Besides the use of statistical approaches for terminology identification, there are also the symbolic approaches based on syntactic description of terms. Hu & Liu [Hu and Liu 2004b] adopted the symbolic approach for ATE with a subtle use of statistics. They specifically look for noun phrases, which usually produce a high recall, limiting their number by their frequency of occurrence, which is the basis for pruning the term possibilities. De Clercq and colleagues [De Clercq et al. 2015] have also applied a hybrid ATE strategy. They built a system called TExSIS. The system applies a POS tagger, a lemmatizer and a chunker. All the nouns and noun phrases are considered candidate terms. Then, they use statistical filters to determine the specificity of and cohesion between these candidate terms.

A more closely related work to the one reported here has been the one of Pavlopoulos & Androutsopoulos [Pavlopoulos and Androutsopoulos 2014]. Their work focus is ATE, but it comes as one of the steps towards their research for aspect-based sentiment analysis (ABSA). To perform ATE, the authors applied both, a statistical (FREQ Baseline) and a symbolic method (Hu & Liu), and they have also used the distributional approach for word representation W2V [Mikolov et al. 2013] to prune these prior two techniques. This paper differs from the previous one in two points: (i) we applied the W2V method not only as a pruning strategy after the Hu & Liu method, but also for the resulting infrequent terms; and (ii) we have tested this sequence of ATE methods for book reviews in Portuguese, which shows to be a very challenging domain.

A survey on ATE for the Portuguese language [da Silva Conrado et al. 2014] has shown few NLP research groups are investigating the task of term extraction for Portuguese [Lopes et al. 2009, Lopes et al. 2010, da Silva Conrado et al. 2013, Conrado et al. 2013] and none of them has used the approach that is presented here, which is detailed in what follows.

3. Material and methods

3.1. The dataset

All the methods cited before will be detailed below and have been applied to the ReLi corpus [Freitas et al. 2014], a Portuguese book review corpus consisting of 1,600 reviews for 14 distinctive books, containing 12,470 Portuguese sentences and 259,978 words. ReLi had its sentences manually tagged by linguistic experts for their aspects and opinions/polarities. Among these sentences, there are those which express opinions about either a book or a particular aspect of the book. There are also neutral sentences, without polarity. Overall, there are 2,883 positive opinion sentences, 596 negative opinion sentences and 212 neutral sentences.

The evaluation of the methods was performed by comparing the returned targets (the identified aspects) to the corresponding reference/gold targets annotated in the corpus. The evaluation methodology presented in this article is similar to the one used for the SemEval workshops (see, e.g., [Pontiki et al. 2015]). The targets were extracted with their respective starting and ending offsets locations in the sentence. Only distinct targets were considered, and NULL targets were discarded. Precision and recall were computed, using a micro-averaging method [Sebastiani 2002]. F-measure, a harmonic mean of precision and recall, were also computed.

Considering a contingency table and the classic information retrieval notions of precision (P) and recall (R), an evaluation measure is based on the number of true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn). In this way, micro-averaging P^μ and micro-recall R^μ may be achieved by summing all individual decisions, such as:

$$P^\mu = \frac{\sum_{c=1}^C \text{tp}_i}{\sum_{c=1}^C (\text{tp}_i + \text{fp}_i)} \quad R^\mu = \frac{\sum_{c=1}^C \text{tp}_i}{\sum_{c=1}^C (\text{tp}_i + \text{fn}_i)}$$

where C is the number of c classes under evaluation and i represents individual instances in these classes.

3.2. The methods

We implemented and evaluated three approaches for ATE, which are summarized below.

FREQ Baseline This ATE method (referred simply by `FREQ` here) is perhaps the simplest form of term identification. It is entirely based on the frequency of occurrence of nouns and noun phrases. It returns the most frequent nouns and noun phrases in a text after a pruning mechanism, keeping only the words with a certain frequency to limit the eventual error of labeling every noun or noun phrase as a relevant term. This method has been reported as very effective by various researchers [Hu and Liu 2004a, Wei et al. 2010, Liu 2012].

Hu & Liu This method may be categorized as a hybrid term detection method because it is based on the frequency strategy over tokens previously tagged. This method has the following advantages over the `FREQ` method: it detects longer and compound terms; it may detect infrequent terms; and it tries to correct itself by pruning these incorrect terms.

The Hu & Liu algorithm may be summarized in eight steps, synthesized below:

1. Frequent term detection: nouns and noun phrases should be extracted from sentences using a POS tagger and a parser. This method ignores terms below a frequency less than 1% in the corpus. In our implementation, we vary this frequency between 1 and 10% to choose a more suitable value.
2. Identification of compound noun terms: noun bigrams and triples of nouns in a sentence shall be extracted in the same order they appear. The resulting list of nouns and noun phrases includes the *candidate terms*. For example: the word ‘campo’ may appear many times in a text, but this word comes always inside the expression “O apanhador no campo de centeio”. This way, the triple ‘apanhador campo centeio’ will be extracted as a compound candidate term.
3. p -support (pure support) calculation: the p -support of a candidate term t is the number of sentences containing t , excluding sentences where another candidate t' appears and the t candidate is a subset of t' .
4. Compactness score calculation: in a sentence that contains a compound candidate term, the distance in words between each pair of terms of the candidate is verified. If this distance is greater than 3 words, the score is increased.
5. Compactness pruning: term candidates that appear more than once in a non-compact form are removed.
6. p -support pruning: candidates with a p -support value less than three are also removed.
7. Extraction of Opinion words: all the adjectives that appear close to a term candidate shall be extracted.
8. Infrequent term detection: all sentences without any term are reanalyzed and, if an opinion word is found, its closest noun is added to the list of candidate terms.

We have also applied the Word2Vec model. It was used as a subsidiary pruning method to the previous `FREQ` and Hu & Liu methods. The W2V model was designed by Mikolov and his colleagues at Google [Mikolov et al. 2013] as an efficient method for learning word vector representations from large corpora. The W2V method is based on an earlier distributed representation of words, the vector space model (VSM), in which words are represented as dense vectors in a continuous vector space. In the W2V model, the vectors are semantic representations of words and are treated as having latent variables, each one representing a “dimension” of the word meaning. In this representation, semantically similar words have vector representations close to one another. One may also say that semantically similar words are *embedded nearby each other*. Before W2V, the adopted Distributional Hypothesis was based on the statistics of how often some word co-occurred with its neighbor words in a large text corpus. In a different way, W2V uses a predictive model, a computationally efficient predictive neural probabilistic language model. Predictive models directly attempt to predict a word from its neighbors in terms of learned embedded vectors.

Formally, neural probabilistic language models are trained using the maximum likelihood principle to maximize the probability of the next word, w_{t+j} , in a neighborhood $|c|$, given the previous words w_j . Given a sequence of training words $(w_1, w_2, w_3, \dots, w_T)$ in a corpus with T words, the objective of the W2V model is to maximize the following

average log probability:

$$J_{ML} = \frac{1}{T} \sum_{t=1}^T \sum_{-c < j < c, j \neq 0} \log p(w_{t+j} | w_j)$$

In the above formulation, c refers to the size of the training context for the center word w_t . Larger values of c result in more training samples, which, as a consequence, may lead to a higher accuracy model. In other words, the aim of this method is to try to correlate words in a vocabulary by measuring their compatibility via probability values.

Each term vector may be constructed by training a probabilistic language model from a corpus. For Portuguese texts, we used the model trained by Rodrigues and his colleagues [Rodrigues et al. 2003], using data from Wikipedia, scientific and journalistic texts, lectures, movie subtitles, technical texts, and other documents.

As the W2V approach is used for pruning unrelated term lists, it was necessary to calculate a couple of specific term vectors, known as the “context” vector and the “common” vector. The former is a term vector extracted from the top 10 most frequent terms of the corpus under analysis. It is meant to be a representative vector of the context information in the ReLi corpus. The latter is a term vector created from the top 20 most frequent terms in a corpora representing the Portuguese language as a whole. Its purpose is to serve as a representative vector of the common terms of a language. Both, the context and the common vectors, are centroid vectors. The distinction in the number of most frequent terms to create both term vectors is based on the intuition that the candidate terms are surrounded by more general concept terms than the domain-specific terms. In order to know if one term belongs to the book context, its representative vector must be compared to both vectors using the traditional cosine similarity. If the result is closer to the context vector the term is maintained, if it is closer to the common vector the term is discarded.

FREQ Baseline + W2V Pruning In the FREQ model for single terms, words are treated as discrete atomic symbols. In a different direction, the W2V model steers words towards their meanings. We believe that, using the W2V model, the FREQ method may have its threshold pruning step extended or even enhanced.

Besides the frequency pruning of the FREQ method, an extra pruning may be used to further reduce the number of non-related detected terms. This extra pruning step is sustained by the concept of *term vector*, which is a vector representative of a term on its context. The terms found by the FREQ Baseline method were pruned based on their distance to each context vector, as explained before.

Hu & Liu + W2V Pruning We have also modified the Hu & Liu method to add W2V pruning. This method has already a standard pruning step in its first stage, called *Frequent term detection*. After this first pruning, the W2V method was also applied as suggested by [Pavlopoulos and Androutsopoulos 2014]. With this second pruning, the intention was to remove the non-context terms.

Hu & Liu + W2V Pruning + Infrequent term processing with W2V For each term located near an opinion word, a W2V analysis is performed. If the term belongs to the context under study (books), it is added to the infrequent term list; otherwise, the algorithm moves on to the analysis of the next closest term. This process is repeated until a contextualized term is found and, if it happens that no term is found, the algorithm considers as infrequent term the one closest to the opinion word.

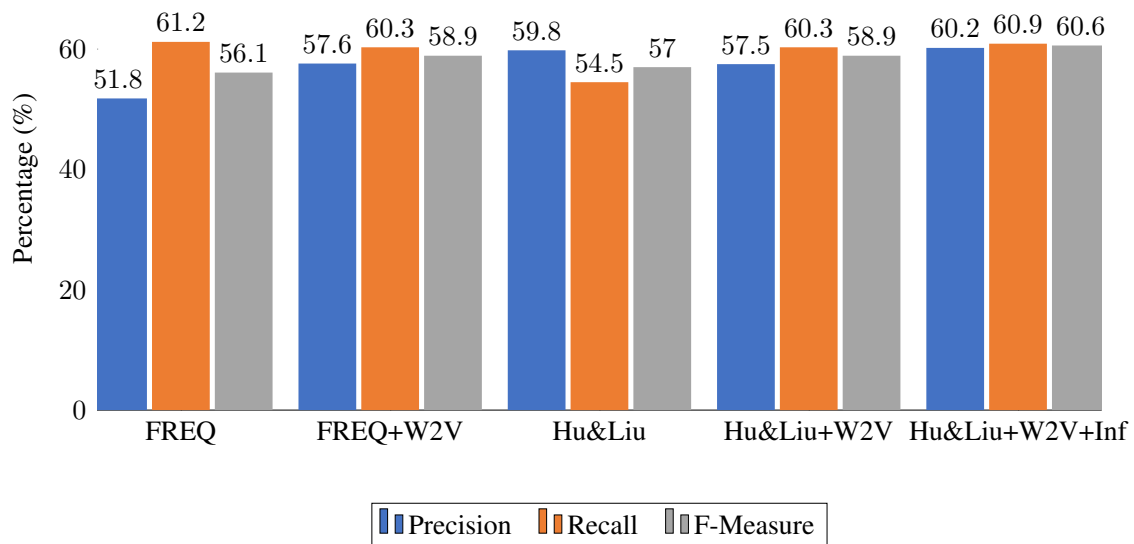
4. Results

We analyzed the behavior of the algorithms in two situations: a) with a relatively small dataset containing 2,233 opinion phrases from ReLi corpus, and b) with an extended dataset with additional 51,149 sentences obtained from book reviews collected from the skoob¹ site, the same site that originated the ReLi corpus.

It is worth recalling that, as a preliminary step, both corpora have been submitted to a pre-processing phase, when a tokenizer [Bird et al. 2009] was applied at the sentence level. For part-of-speech tagging, the `nlpnet` POS tagger [Fonseca and Rosa 2013] was applied.

The results obtained for the statistical, symbolic and distributional methods are shown in Figure 1 for the ReLi Corpus and in Figure 2 for the Extended ReLi Copus. A detailed description of how the methods were applied and their results are presented below.

Figure 1. ReLi Corpus results.

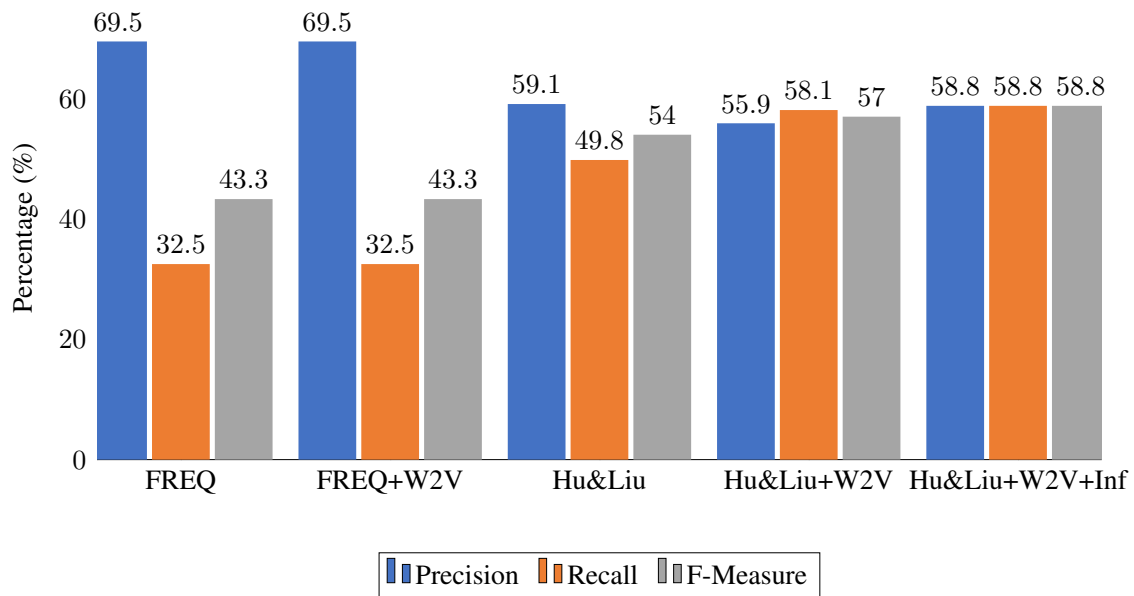


4.1. FREQ Baseline

All nouns found on the pre-processing phase were organized in descending order by their frequencies. All terms below a certain frequency were excluded. To identify this cutoff

¹<https://www.skoob.com.br/>

Figure 2. Extended ReLi Corpus results.



frequency, the datasets were processed with different values, in a range of 1% to 10% to find the best fit value for the F-measure. Figures 1 and 2 show very contrasting values for both corpora, the standard ReLi and its extended version. While recall showed better results for the ReLi corpus, the same did not happen for the extended ReLi, where we obtained the highest precision value and also the lowest values for recall and f-measure of all the tests. This result may be explained by the fact that the `FREQ` method is purely a frequency based method. When the `FREQ` is executed over a large dataset, the amount of resulting terms tends to be smaller than if processed for a smaller corpus, but the results usually comes with a better quality, thus implying higher values for precision and lower values for recall.

4.2. `FREQ` Baseline + W2V Pruning

The W2V pruning of non-context terms enhanced the precision (from ≈ 51.8 to ≈ 57.6) and f-measure (from ≈ 56.1 to ≈ 58.9) when applied only to the standard ReLi corpus. The words ‘mundo’, ‘tempo’ and ‘forma’ are examples of non-context terms that have been correctly excluded from the final set of detected aspects. With the extended corpus, the pruning step had no effect. See Figures 1 and 2. In a larger corpus, non-context terms were naturally eliminated due to their relative lower frequency, thus nulling the effect of the W2V pruning stage.

4.3. Hu & Liu

The Hu & Liu method has the advantage of detecting compound terms, having three pruning stages (by frequency, compactness, and p-support) and detecting infrequent terms. Words like ‘titulo’ and ‘texto’, that are not usually so frequent, have been now correctly identified. For the ReLi corpus, precision values increased. Although this method is a more sophisticated ATE method compared to the `FREQ` Baseline, some non-context

terms still appeared, which led to a decrease in recall. For the extended ReLi corpus, recall increased but precision had a considerable decrease.

4.4. Hu & Liu + W2V Pruning

The effect of this pruning step is clear in finding terms such as ‘personagem’, ‘Holden’ and ‘obra’. As also seen in Figures 1 and 2, the W2V pruning step enhanced the prior recall results of the stand-alone Hu&Liu and decreased precision values for both corpora. However, the value of F-measure improved, indicating that, in general, the use of this new pruning approach was positive.

4.5. Hu & Liu + W2V Pruning + Infrequent term processing with W2V

Differently from [Pavlopoulos and Androutsopoulos 2014], we tested the W2V pruning in the last step of the Hu & Liu method, on the infrequent term list, as explained before. The improvement with this is step over the infrequent terms may be exemplified for discarding the term ‘justificativa’ which is not a related infrequent aspect. It also correctly chooses ‘texto’ instead of ‘sacadas’ and ‘enredo’ instead of ‘adolescente’. Figures 1 and 2 shows the obtained improvements. The best values were obtained for all the measurements, disregarding the one obtained in the $FREQ$ with the extended ReLi, result explained on Section 4.1.

5. Conclusions and final remarks

Our results show that the tested methods obtain varied performances for the two corpora, the standard ReLi and its extended version. However, one may notice that incorporating the distributional properties of words (the W2V strategy) generally improves the results. Overall, we may see that the distributional approach of W2V has caused an impact in both the statistical $FREQ$ Baseline method and the hybrid Hu & Liu method. Applying the word2vec has increased the precision of these methods up to 10 percent points (pp), showing the advantages of considering distributional semantics in the process.

A great advantage of pruning with W2V is the removal of context unrelated terms. When this removal was applied only to the infrequent term candidates, one may see a general decrease in the measures, showing that most of these infrequent terms are also part of the context under study. The removal of infrequent term candidates after W2V increased the results by a very small margin for the ReLi corpus.

To the interested reader, examples of correctly detected aspects on ReLi corpus are: ‘livro’, ‘história’, ‘narrativa’, ‘personagem’, and ‘escrita’. Some examples of non-detected aspects are ‘o apanhador no campo de centeio’ and ‘adolescente do livro’, which were expressions that appeared very few times.

Acknowledgements

This research was partially supported by FAPESP.

References

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O’Reilly Media, Inc.

- Castellví, M. T. C., Bagot, R. E., and Palatresi, J. V. (2001). Automatic term detection: A review of current systems. *Recent advances in computational terminology*, 2:53–88.
- Conde, A., Larrañaga, M., Arruarte, A., Elorriaga, J. A., and Roth, D. (2016). Litewi: A combined term extraction and entity linking method for eliciting educational ontologies from textbooks. *Journal of the Association for Information Science and Technology*, 67(2):380–399.
- Conrado, M. S., Rossi, R. G., Pardo, T. A. S. P., and Rezende, S. O. (2013). Applying transductive learning for automatic term extraction: The case of the ecology domain. In *2013 Second International Conference on Informatics Applications (ICIA)*, pages 264–269.
- da Silva Conrado, M., Felippo, A. D., Salgueiro Pardo, T. A., and Rezende, S. O. (2014). A survey of automatic term extraction for Brazilian Portuguese. *Journal of the Brazilian Computer Society*, 20(1):12.
- da Silva Conrado, M., Pardo, T. A. S., and Rezende, S. O. (2013). A machine learning approach to automatic term extraction using a rich feature set. In *HLT-NAACL*, pages 16–23.
- De Clercq, O., Van de Kauter, M., Lefever, E., and Hoste, V. (2015). Applying hybrid terminology extraction to aspect-based sentiment analysis. In *International Workshop on Semantic Evaluation (SemEval 2015)*, pages 719–724. Association for Computational Linguistics.
- Fonseca, E. R. and Rosa, J. L. G. (2013). Mac-morpho revisited: Towards robust part-of-speech tagging. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, pages 98–107.
- Frakes, W. B., Kulczycki, G., and Tilley, J. (2014). *A Comparison of Methods for Automatic Term Extraction for Domain Analysis*, pages 269–281. Springer International Publishing, Cham.
- Frantzi, K., Ananiadou, S., and Mima, H. (2000). Automatic recognition of multi-word terms: the C-value/NC-value method. *International Journal on Digital Libraries*, 3(2):115–130.
- Frantzi, K. T. and Ananiadou, S. (1997). Automatic Term Recognition using Contextual Cues. In *In Proceedings of 3rd DELOS Workshop*.
- Freitas, C., Motta, E., Lilidiú, R. L., and César, J. (2014). Sparkling Vampire... lol! Annotating Opinions in a Book Review Corpus. In Aluísio, S. and Tagnin, S. E. O., editors, *New Language Technologies and Linguistic Research: A Two-Way Road*, chapter 8, pages 128–146. Cambridge Scholars Publishing, Cambridge.
- Hu, M. and Liu, B. (2004a). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177. ACM.
- Hu, M. and Liu, B. (2004b). Mining opinion features in customer reviews. In *AAAI*, volume 4, pages 755–760.

- Kageura, K. and Umino, B. (1996). Methods of automatic term recognition: A review. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 3(2):259–289.
- Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.
- Lopes, L., Fernandes, P., Vieira, R., and Fedrizzi, G. (2009). E χ ATOLP—An Automatic Tool for Term Extraction from Portuguese Language Corpora. In *Proceedings of the 4th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 427–431.
- Lopes, L., Oliveira, L. H., and Vieira, R. (2010). Portuguese term extraction methods: Comparing linguistic and statistical approaches. In *International Conference on Computational Processing of Portuguese Language*, volume 6.
- Lossio-Ventura, J. A., Jonquet, C., Roche, M., and Teisseire, M. (2016). Biomedical term extraction: overview and a new methodology. *Information Retrieval Journal*, 19(1-2):59–99.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Nakagawa, H. and Mori, T. (2002). A simple but powerful automatic term extraction method. In *COLING-02 on COMPUTERM 2002: Second International Workshop on Computational Terminology*, volume 14, pages 1–7. Association for Computational Linguistics.
- Pavlopoulos, J. and Androutsopoulos, I. (2014). Aspect term extraction for sentiment analysis: New datasets, new evaluation measures and an improved unsupervised method. *Proceedings of LASMEACL*, pages 44–52.
- Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., and Androutsopoulos, I. (2015). Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.
- Ravi, K. and Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89:14–46.
- Rodrigues, J., Branco, A., Neale, S., and Silva, J. (2003). LX-DSemVectors: Distributional Semantics Models for Portuguese. *6th International Workshop PROPOR'2003, Faro, Portugal, June 2003*, 8775(2721):214–219.
- Schouten, K. and Frasincar, F. (2016). Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.
- Wei, C.-P., Chen, Y.-M., Yang, C.-S., and Yang, C. C. (2010). Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews. *Information Systems and E-Business Management*, 8(2):149–167.

Diagnóstico de benignidade e malignidade em imagens mamográficas utilizando Aprendizado Profundo para extração das regiões de interesse e Índices de Diversidade Filogenéticas para classificação

Ismael de H. Leal¹, kelson Rômulo T. Aires¹, Rodrigo de M. S. Veras¹,

¹Núcleo de Computação de Alto Desempenho – Universidade Federal do Piauí (UFPI)
CEP 64.049-550 – Teresina – PI – Brazil

{ismaelleal30, krtaires, }@gmail.com,

rveras@ufpi.edu.br

Abstract. *Breast cancer is indicated as the main cause of death among women. Its occurrence increases considerably every year. Early diagnosis is directly related to its chances of successful recovery. One way of assisting healthcare professionals in identifying and diagnosing this disease is through Computer Aided Detection/Diagnosis Systems. This research proposes a methodology for the detection of malignity and benignity patterns in mammographic images, by analyzing features extracted from the phylogenetic diversity indexes. The region of interest was extracted with the aid of a Convolutional Neural Network, based on the U-Net network architecture. For classifying and distinguishing benign and malign images, three approaches were used: J48, LMT and RandomForest. This methodology presented positive results in the classification process. The results were compared to those of some related works.*

Resumo. *O câncer de mama é apontado como a principal causa de morte entre as mulheres. Sua ocorrência aumenta consideravelmente todos os anos. O diagnóstico dessa doença de forma precoce está diretamente relacionado às suas chances de cura. Uma das formas de auxílio ao profissional de saúde na identificação e diagnóstico é o uso de Sistemas de Detecção e Diagnósticos auxiliados por computador (Computer Aided Detection/Diagnosis). Este trabalho propõe uma metodologia para discriminação de padrões de malignidade e benignidade em imagens mamográficas, através das características extraídas pelos índices de diversidade filogenética. A região de interesse foi extraída através da rede neural artificial CNN (Convolutional Neural Network), com base na arquitetura da rede U-Net. Para a classificação, visando diferenciar as imagens malignas e benignas, utilizaram-se três abordagens: J48, LMT e o RandomForest. A metodologia apresentou bons resultados para a classificação. O resultado foi comparado com alguns trabalhos relacionados.*

Introdução

Segundo o Instituto Nacional do Câncer (INCA), o câncer de mama é o mais comum entre as mulheres no mundo. No Brasil, estima-se que cerca de 596.070 novos casos para o ano de 2017, sendo que 57.960 serão de câncer de mama [INCA 2017].

A relação entre tratamento e cura do câncer de mama está diretamente relacionada à sua detecção precoce. A indicação é que mulheres entre 50 e 69 anos façam exames periódicos a cada dois anos. Um dos exames mais indicados é a mamografia.

A avaliação do exame mamográfico exige grande conhecimento do especialista para detectar possíveis anomalias ou patologias, podendo este ser relacionado a nódulos malignos e benignos. Técnicas computacionais têm se tornado cada vez mais usadas para auxílio dos especialistas. Os sistemas de diagnósticos de doenças auxiliados por computador (*CADx – Computer-Aided Diagnosis*) são ferramentas que utilizam os recursos computacionais para fornecer mais informações ao especialista no diagnóstico, diminuindo possíveis erros.

O processamento digital de imagens estuda requisitos para extrair informações técnicas, a fim de melhorar os parâmetros para um diagnóstico mais preciso, aumentando a interpretação da mamografia [Silva et al. 2016]. A análise de texturas é um dos requisitos para extrair informações. As mamografias fornecem uma descrição dos pixels do tecido mamário, onde cada textura possui um padrão de características.

O trabalho tem como objetivo apresentar uma nova metodologia para classificação de texturas da mama em malignos e benignos, usando rede neural artificial para extrair a região de interesse e índices de diversidade filogenética para extração das características. Em seguida, foi feita a identificação com alguns classificadores.

Este trabalho está dividido da seguinte forma: A seção 2, são apresentados os trabalhos relacionados; na seção 3 está descrita os materiais e métodos; na seção 4, temos os resultados obtidos e a discussão sobre o mesmo; e por fim, na seção 5, temos a conclusão

Trabalhos Relacionados

Existem na literatura muitos trabalhos realizados para a extração da região de interesse e classificação de imagens da mama em malignos e benignos, em exames de mamografias. As características extraídas das imagens contendo as regiões de interesse servem de vetores de entrada para os classificadores.

[Matos et al. 2017] apresenta uma metodologia que combina detectores e descritores de características locais, com um modelo de representação de dados para análise, tanto de textura quanto de geometria em regiões extraídas das mamografias. São utilizados detectores SIFT, SURF e ORB, e descritores HOG, LBP, BRIEF e Haar Wavelet. A classificação utilizou três abordagens: *Support Vector Machine*, *RandomForest* e *Adaptive Boosting*. Após os teste, obteve como melhor resultado o classificador *RandomForest* com acurácia de 80,23%, sensibilidade de 89,57% e especificidade de 69,18%.

[Silva et al. 2016] apresenta uma abordagem para classificação de massas em maligno e benigno, a qual utiliza a base DDSM (*Digital Database for Screening Mammography*). As características são extraídas utilizando as técnicas de *Local Binary Pattern* (LBP), Função K de *Ripley* e os índices de *Shannon*, *Mcintosh*, *Simpson*, *Gleason* e de *Menhinick*, para extração das características de formas de imagens de mamografias. Para classificação utilizou a Máquina de Vetor de Suporte (SVM). Após os testes, obteve como resultado uma acurácia de 90,18%, sensibilidade de 91,01% e especificidade de 89,94% para o grupo com todas as imagens.

[Rocha et al. 2014] usa LBP, Geoestatística e Índice de Diversidade para extração

das características de textura das imagens de mamografias da base DDSM para classificação das massas em maligno e benigno. Para classificação, utilizou a Máquina de Vetor de Suporte (SVM). O melhor resultado obtido com os testes foram: acurácia de 92,20%, sensibilidade de 92,26% e especificidade de 91,26% e área sobre a curva ROC de 0,920.

Diante dos trabalhos mostrados, pode-se perceber a utilização de vários métodos de extração de características, como também a utilização de vários classificadores. Este trabalho tem como objetivo extrair as *Regions of Interest* (ROI's), utilizando uma rede neural CNN (*Conventional Neural Network*), com base na arquitetura da rede *U-Net* para classificação em maligno e benigno baseado em Índices de Diversidade Filogenética.

Materiais e métodos

Para que a classificação das imagens de mamografias malignas e benignas fosse possível, foi empregada a seguinte metodologia: Primeiro foi feita a aquisição das imagens oriundas da base *Digital Database for Screening Mammography* (DDSM). Segundo foi utilizado *Deep Learning* com uma *Convolutional Neural Network* (CNN), com arquitetura da rede *U-Net* para a extração da região de interesse (ROI). Em seguida, foi realizada a etapa de extração de características das ROI's, utilizando descritores de Índices de Diversidade Filogenética baseados na textura. Para a classificação foram utilizados os classificadores *RandomForest*, J48 e LMT.

Anatomia e patologia da mama

A mama, também conhecida como glândula mamária, é composta por tecido conectivo, adiposo e tecido mamário, que contém glândulas que produzem leite. A glândula mamária é composta por lobos e dutos (tubos por onde o leite passa). Cada mama apresenta entre 15 a 20 lobos, que contém pequenos lóbulos. Os lóbulos têm nas duas extremidades dezenas de pequenos bulbos com finalidade de produzir leite [Bochartt et al. 2013].

Uma neoplasia (ou tumor) é uma proliferação anormal de células do corpo. Surge da mutação genética onde uma ou mais células sofrem alteração no DNA e perdem sua função no tecido que fazem parte e passam a se reproduzir descontroladamente [Resmini et al. 2011]. A neoplasia benigna tem formato definido e não agride os tecidos vizinhos, além de crescer lentamente. A neoplasia maligna é mais disforme, possuindo mais ramificações, se alimenta das células vizinhas e aumenta rapidamente.

O mamógrafo é o aparelho usado para realização do exame da mamografia. Ele emite uma radiação ionizante (feixes de Raios-X) sobre a mama comprimida. Os Raios-X emitidos contra a mama são retidos em partes pelos tecidos mamários. Quanto maior a densidade da mama, maior será a quantidade de raios-x retidos pelo tecido, dificultando assim a visualização de possíveis calcificações.

Base

A base de imagens DDMS é uma base pública contendo imagens de mamografias, que tem como objetivo facilitar a pesquisa e desenvolvimento de algoritmos para ajudar no diagnóstico de anomalias da mama. A base foi uma doação do *Breast Cancer Research Programo of the U.S. Army Medical Reserch and Materiel Comman*. O banco possui 2500 estudos. Cada estudo contém duas imagens de cada mama, juntamente com informações

do paciente, tais como classificação da densidade da mama, sutileza, para anomalias e informações de imagens, como resolução espacial e imagens contendo áreas suspeitas [Health et al. 2000]. Cada estudo possui um *Overlay* que é a marcação das massas nas mamas feitas por um especialista.

Deep learning e Convolutional Neural Network

Machine Learning (ML), são técnicas que utilizam algoritmos capazes de aprender uma representação compacta dos dados, prever ou generalizar padrões para dados desconhecidos, a partir de um conjunto de treino conhecido. A rede neural artificial do tipo *Deep Learning* é uma das técnicas de ML mais recentes para analisar grandes quantidades de dados e extrair conhecimento dos mesmos. A capacidade de aprendizagem de forma não-supervisionada em grandes quantidades de dados torna a principal vantagem dos algoritmos do tipo *Deep Learning*.

Os algoritmos DL apresentam uma abordagem inovadora ao gerar, de forma automática, propriedades invariantes nas suas camadas de representação hierárquicas. Estes métodos têm produzido excelentes resultados em diferentes aplicações, incluindo o reconhecimento de voz, visão computacional, e processamento de linguagem natural [Goodfellow et al.]. Estes métodos conseguem obter uma representação complexa e abstrata de forma hierárquica usando várias camadas de processamento não-linear. Os dados sensoriais (pixels de uma imagem, por exemplo) são alimentados numa primeira camada, sendo a saída de cada camada uma entrada da camada seguinte.

Uma das redes neurais que mais se destaca é a *Convolutional Neural Network* (CNN), que é o conjunto de múltiplas tarefas de aprendizado de máquina e tem como característica mais marcante a resolução de problemas a partir de estruturas de representação dos dados de diferentes maneiras. Consistem em um conjunto de camadas que extraem características de imagens de entrada, fazendo sucessivas convoluções e redimensionamento, de modo que no final se consegue apenas ficar com a marca da classe quem a imagem pertence [Neves et al. 2011].

A arquitetura de CNN utilizada neste trabalho é a *U-Net: Convolutional Networks for Biomedical Image Segmentation* [?]. É uma arquitetura de rede convolucional para segmentação rápida e precisa de imagens. Essa rede foi utilizada primeiramente para a segmentação de estruturas neuronais em pilhas de microscópio eletrônico, chegando a ganhar o desafio *Grand Challenge for Computer-Automated Detection of Caries in Bitewing Radiography*, no ISBI 2015 . Seu funcionamento está ilustrado na Figura 1. Na figura, podemos visualizar uma arquitetura de uma unidade. Como todas as outras redes convolucionais, ela consiste em um grande número de operações diferentes ilustradas pelas pequenas setas. A imagem de entrada é alimentada na rede no início. Em seguida, os dados são propagados através da rede ao longo de todos os caminhos possíveis e, em seguida, no final a rede de segmentação sai pronta.

Cada caixa azul corresponde a um recurso multicanal. O tamanho x-y é indicado pelo número que se encontra na parte inferior da camada e os canais de recursos são indicados pelos números que ficam na parte superior da camada. A maioria das operações são convoluções seguidas de uma função de ativação não-linear. O projeto usa somente a parte desenvolvida da convolução, isso significa que para uma convolução de três, um pixel é perdido. Isso permite processar imagens grandes em títulos individuais. A operação

seguinte é a operação max pool, ela reduz o tamanho x-y da rede futura, ilustrado na Figura 1, como uma seta para baixo.

A operação max-pooling atua em cada canal separadamente. Ele propaga a ativação máxima de cada janela de dois por dois para o próximo mapa de recursos. Após cada operação de max-pooling, é aumentado o número de canais de recurso por um fator de dois. A rede de classificação padrão termina no canal de recurso com tamanho 1024 e mapeia todos os recursos para um único recurso de saída. A unidade tem um caminho de expansão adicional para criar um mapa de segmentação de alta resolução. [Ronneberger et al. 2015]

O principal desafio da rede, é a segmentação de imagens com baixo número de dados para treinamento. Para ensinar a rede robustas propriedades, aplica-se deformações elásticas aleatórias. A imagem deformada resultante parece perfeitamente como a imagem original.

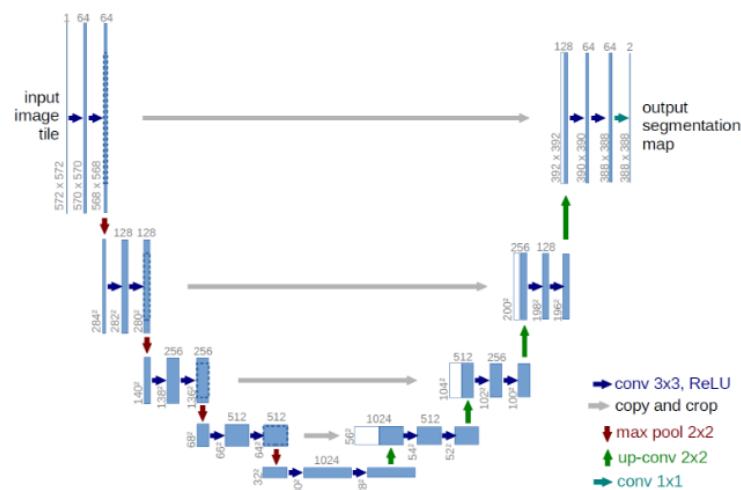


Figure 1. Arquitetura da rede U-Net. FONTE: [Ronneberger et al. 2015]

Treinamento e aplicação da rede *U-Net*

Para o desenvolvimento deste trabalho utilizou-se 2187 imagens da base DDSM. Para cada imagem utilizada, existe uma imagem correspondente com um *overlay* indicando a marcação das massas por um especialista. A partir das imagens do *overlay* com a marcação, foi criado um script que extrai apenas a ROI das imagens sem a marcação, tornando assim, a base de treinamento da rede um conjunto de imagens contendo 4.374 imagens, sendo 2187 imagens da mama e 2187 imagens da máscara correspondente. As máscaras apontam as características a serem aprendidas para a detecção de massas para uma futura classificação, que se diferenciam entre benignas ou malignas.

A base contendo 4.374 imagens foi definida como dados de entrada para início do treinamento da rede para reconhecimento de padrões. A CNN baseada na rede *U-Net* foi treinada algumas vezes com a finalidade de se achar o melhor resultado para aprendizagem. Com um total de 10 convoluções, o *batch_size* de tamanho 32 e com o número de época igual a 20, conseguiu-se o melhor coeficiente de aprendizagem com 76,71%.

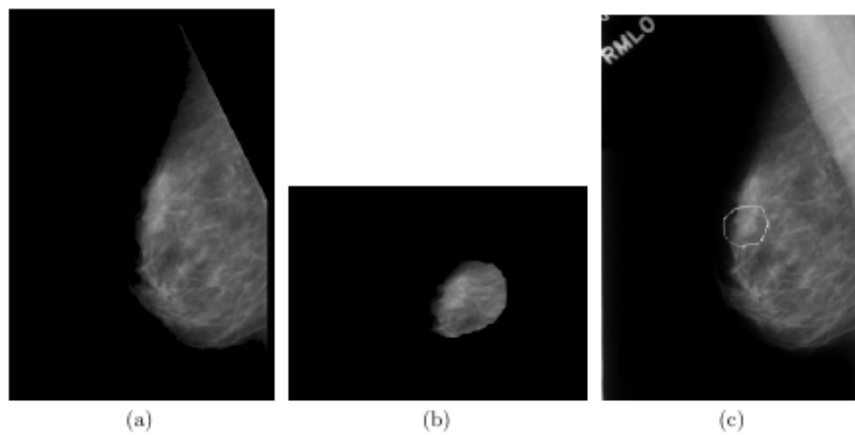


Figure 2. Detecção da Região de Interesse

Após o treinamento da rede, foram salvos os pesos que continham as características e padrões aprendidos sobre as massas das imagens da mama. Com isso, foram utilizadas 200 imagens, escolhidas aleatoriamente para testes, sendo que 100 eram imagens diagnosticadas com massas benignas e 100 contendo massas malignas. Como resultado, a rede identificou as ROI's de cada imagem contendo as massas para posterior classificação. Como podemos ver nas imagens da figura 2.

A Figura 2 nos mostra a imagem (a) como uma imagem escolhida de forma aleatória para servir de teste da rede neural convolucional. A imagem (b) apresenta a região de interesse segmentada pela Rede U-net da imagem (a). A imagem (c) nos mostra a mesma imagem (a), mas com a marcação do especialista para fim de comparação entre a imagem extraída e a imagem com o *overlay*.

O passo seguinte foi utilizar descritores de texturas baseados em Índices de Diversidade Filogenética nas ROI's resultadas do processamento da rede, para que pudesse ser feita a classificação entre malignas e benignas.

Índices de Diversidade Filogenética

Filogenia é um ramo da biologia responsável pelo estudo das relações evolutivas entre as espécies. Seu objetivo é informar a variedade de espécies presentes em uma comunidade ou área [Webb et al. 2000]. Um dos objetivos de usar a filogenia é averiguar a composição das espécies nas comunidades de forma contemporânea junto aos processos ecológicos. A forma mais simples da aplicação do índice de diversidade em imagens é feita quando a comunidade representa uma imagem ou região da mesma, as espécies sendo os níveis de cinza, os indivíduos sendo os pixels e as distâncias filogenéticas, sendo os números de arestas entre duas espécies [Oliveira et al. 2013].

Neste trabalho são utilizados cinco índices de diversidade filogenética para extração dos descritores de textura: *Phylogenetic Diversity* (PD), *Sum of Phylogenetic Distances* (SPD), *Mean Nearest Neighbour Distance* (MNND), *Phylogenetic Species Variability* (PSV) e *Phylogenetic Species Richness* (PSR). O índice de diversidade filogenética PD é o somatório dos comprimentos dos ramos da filogenia de cada espécie, onde B é o número de ramificações da árvore, Li é o comprimento do ramo e Ai é a abundância média de espécies que compartilham o ramo i , conforme mostra a Equação 1.

$$B \times \frac{\sum_i^B L_i A_i}{\sum_i^B A_i} \quad (1)$$

O índice de diversidade filogenética SPD é a soma das distâncias filogenéticas entre cada par de espécies, onde d é a distância entre as espécies m e n ; a_m é a abundância de espécies m ; S é o número de espécie do conjunto focal. Mostrado na equação 2:

$$\left(\frac{S(S-1)}{2} \right) \times \frac{\sum \sum_{m < n} d_{mn} a_m a_n}{\sum \sum_{m < n} a_m a_n} \quad (2)$$

O índice de diversidade filogenética MNND é a distância média do táxon mais próximo, onde d é a distância entre as espécies m e n ; a_m é a abundância de espécies m . Conforme mostra a equação 3:

$$\sum_m^S \min(d_{mn}) a_m \quad (3)$$

O índice de diversidade filogenética PSV é a variedade de espécies filogenéticas que resume o grau em que as espécies em uma comunidade são filogeneticamente relacionadas, onde o trC representa a soma dos valores da diagonal de uma matriz C , $\sum c$ é o somatório de todos os valores de uma matriz, n é o número de espécies e c é a média dos elementos da diagonal C . Definidos na equação 4:

$$PSV = \frac{n \, trC - \sum c}{n(n-1)} = 1 - 'c \quad (4)$$

O índice de filogenética PSR é o que quantifica o número de espécies em uma comunidade. Encontra-se o valor do PSR multiplicando o número de espécies n pela variabilidade da comunidade. Como mostra a equação 5.

$$PSR = nPSV \quad (5)$$

Classificação

A classificação foi feita pelo WEKA, que é uma ferramenta utilizada para pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização. A classificação também é adequada para o desenvolvimento de novos esquemas de Aprendizagem de Máquina.

Foram utilizados os classificadores *RandomForest*, J48 e o LMT devido a utilização de descritores baseados em comportamentos dentro de uma comunidade. Todos os classificadores foram utilizados com os valores padrões juntamente com a validação cruzada de k-folds, sendo $k = 10$. Este método divide as características em 10 grupos, de forma a realizar o treino em 9 grupos e utilizando um grupo para testes. São realizados 10 cruzamentos, sempre trocando o grupo de teste que ao final é gerado uma média, onde se configura o resultado.

Validação dos resultados

Para a validação dos resultados sobre a presença ou ausência de massas malignas ou benignas em imagens de mamografias, utilizou-se métricas de avaliação baseadas em estatísticas como: Sensibilidade (S), Especificidade (E) e Acurácia (A).

A matriz de confusão oferece uma hipótese das medidas efetivas do modelo de classificação, mostrando o número de classificações corretas versus as classificações preditas para cada classe sobre um determinado conjunto de exemplo.

A sensibilidade (S) é a proporção de verdadeiros positivos, ou seja, é a capacidade do sistema em identificar corretamente a condição para casos que realmente têm. Como mostra equação 6:

$$S = \frac{VP}{VP + FN} \quad (6)$$

A especificidade é a proporção de verdadeiros negativos, ou seja, é a capacidade do sistema em identificar corretamente a ausência da condição para casos que realmente não a têm. Como mostra a equação 7:

$$E = \frac{VN}{VN + FP} \quad (7)$$

A acurácia é a proporção de predições corretas, sem levar em consideração o que é positivo e o que é negativo. Como mostra a equação 8:

$$A = \frac{VP + VN}{VP + VN + FP + FN} \quad (8)$$

A curva *Receiver Operating Characteristic* (ROC) representa a dependência entre a sensibilidade e a especificidade de um classificador [Rocha et al. 2014]. É um gráfico de sensibilidade (ou taxa de verdadeiros positivos) versus taxa de falsos positivos, que tem como métrica avaliar e comparar o desempenho de duas ou mais modalidades de imagens. Permite evidenciar os valores para os quais existe maior otimização da sensibilidade em função da especificidade.

Resultado e discussões

Para a realização deste trabalho, utilizou-se a base de imagens DDSM, descrita na subseção 3.2. A região de interesse foi extraída utilizando uma rede neural CNN, com arquitetura da rede *U-Net*, onde seu treinamento e aplicação foram descritos na subseção 3.4. A classificação das ROI's entre malignas e benignas foi feita utilizando os classificadores com parâmetros definidos na subseção 3.6.

Para uma melhor observação dos resultados, cada índice de diversidade filogenética foi classificado de forma separada. Inicialmente o índice PD, descrito na seção 3.5, obteve os seguintes resultados:

Analisando a tabela 1, constatamos uma aproximação entre os resultados dos três classificadores, mas que o *RandomForest* conseguiu classificar de uma forma levemente melhor.

Table 1. Tabela 1. Resultados da classificação do índice PD.

Classificador	A(%)	S(%)	E(%)	Curva ROC
J48	58	56,6	60,3	0,552
LMT	58	56,7	60,0	0,574
RandonForest	61,5	61,9	61,2	0,645

O segundo índice classificado de forma separada foi o SPD que é a soma das distâncias filogenéticas entre cada par de espécies, conforme descrito na equação 2. A classificação obteve os seguintes resultados:

Table 2. Tabela 2. Resultados da classificação do índice SPD.

Classificador	A(%)	S(%)	E(%)	Curva ROC
J48	57,5	55,9	60,3	0,560
LMT	56,5	54,9	59,7	0,550
RandonForest	52	52	52	0,573

Observando a tabela 2, percebemos que existe também uma aproximação entres os resultados. O destaque é para o classificador J48, que conseguiu o melhor resultado entre as quatro métricas utilizadas. O *RandomForest* teve o pior desempenho entre todos os classificadores.

O índice MNND, descrito na seção 3.5, ao ser classificado isoladamente, obteve os seguintes resultados, conforme observamos na tabela 3.

Table 3. Tabela 3. Resultados da classificação do índice MNND

Classificador	A(%)	S(%)	E(%)	Curva ROC
J48	52	51	83,3	0,513
LMT	50,5	50,3	50,9	0,501
RandonForest	9	6,4	11,3	0,134

Os resultados observados na tabela 3 mostram uma semelhança entres os classificadores J48 e LMT, onde a especificidade do J48 se destaca de forma muito superior aos demais com 83,3%. Notamos também que o *RandomForest* teve um desempenho muito ruim se comparado aos demais. Sua acurácia foi de apenas 9%.

O índice de diversidade filogenética PSV, que é a variedade de espécies filogenéticas que resume o grau em que as espécies em uma comunidade são filogeneticamente relacionadas. Está descrita matematicamente pela equação 4 na subseção 3.5. Ao ser analisada de forma isolada, obteve os seguintes resultados descritos na tabela 4.

Entre todos os índices analisados de forma separada, o PSV foi o que mostrou melhor resultado. Os classificadores J48 e LMT tobtiveram uma melhor acurácia, sensibilidade e especificidade em relação ao *RandomForest*, mas em compensação, o mesmo obteve um melhor valor sobre a Curva ROC.

Table 4. Tabela 4. Resultados da classificação do índice PSV

Classificador	A(%)	S(%)	E(%)	Curva ROC
J48	87,5	89,5	85,7	0,890
LMT	87,0	86,3	87,6	0,907
RandonForest	85,0	87,6	85,4	0,921

O índice de filogenética PSR, que quantifica o número de espécies em uma comunidade, como mostra a equação 5. Ao ser analisado separadamente, obteve os resultados, como exposto na tabela 5.

Table 5. Tabela 5. Resultados da classificação do índice PSR

Classificador	A(%)	S(%)	E(%)	Curva ROC
J48C	50	50	-	50
LMT	49	49,4	46,4	0,447
RandonForest	46,5	46,3	46,7	0,419

Os classificadores LMT e *RandomForest* tiveram resultados semelhantes, com uma pequena vantagem para o classificador LMT. O classificador J48 não conseguiu fazer a classificação do índice PSR de forma isolada.

Após analisados os índices de forma separada, a classificação foi executada com todos os índices juntos. O padrão utilizado pelos classificadores foi o *Cross-validation* de *k-folds*, sendo $k = 10$. Este método divide as características em 10 grupos, de forma a realizar o treino em 9 grupos e utilizando um grupo para testes. São realizados 10 cruzamentos sempre trocando o grupo de teste que ao final é gerado uma média, onde se configura o resultado. Sendo assim, quanto maior o número de atributos ou características, mais dados serão analisados pelos classificadores, e conseqüentemente, os resultados serão mais confiáveis. Ao juntar todos os índices para a classificação, nota-se uma melhora no desempenho dos classificadores. Os resultados da classificação estão descritos na tabela 6.

Observa-se que o classificador *RandomForest* obteve uma melhor taxa de acertos com 90,5%, uma sensibilidade com 92,6%, uma especificidade com 88,6% e uma curva ROC de 96,7%. Os classificadores J48 e LMT tiveram resultados semelhantes, com uma pequena vantagem para o LMT. Entre os três classificadores, a especificidade do LMT se sobressaiu aos demais, seu resultado foi de 90,7%. Isso mostra que o classificador LMT tem uma capacidade maior em identificar corretamente a ausência da condição para casos que realmente não a têm. Já o *RandomForest* obteve uma sensibilidade muito acima dos demais, atingindo 92,6%. Isso prova que o mesmo tem a capacidade do sistema em identificar corretamente a condição para casos que realmente tem. A curva ROC com resultado de 0,967 mostra que o classificador *RandomForest* consegue detectar um número maior de casos positivos em relação aos classificadores J48 e LMT. Por ser um classificador robusto à presença de ruídos [Breiman 2001], o *RandomForest* apresentou baixa taxa de erros possibilitando um bom resultado na classificação.

Table 6. Tabela6. Resultados da classificação com todos os índices

Classificador	A(%)	S(%)	E(%)	Curva ROC
J48C	88,5	88,1	88,9	0,910
LMT	89,5	88,3	90,7	0,931
RandomForest	90,5	92,6	88,6	0,967

O fato das imagens das mamas contendo diagnósticos malignos e benignos possuírem características muito parecidas, faz com que os dados sejam classificados em grupos diferentes. Observando a tabela 2, podemos verificar que o classificador J48 identificou apenas 88,5% de predição correta.

Table 7. Comparação da metodologia com os trabalhos relacionados

Classificador	A(%)	S(%)	E(%)	Curva ROC
Matos [2017]	80,23	89,57	69,18	-
Silva [2016]	90,18	91,01	89,94	0,960
Rocha [2014]	92,20	92,26	91,26	0,920
Metodologia Proposta	90,5	92,6	88,6	0,967

Observando a tabela 3, o trabalho proposto neste artigo, utilizando índices de diversidade filogenética PD, SPD, MNND, PSV e PSR, apresenta resultados bem significativos em relação ao trabalho de [Matos, 2017]. A diferença superior a 10% na acurácia, mostra uma melhor precisão na detecção dos casos. Com uma especificidade de aproximadamente 20% superior ao trabalho de Matos [2017], o trabalho proposto tem uma capacidade de identificar corretamente a ausência da condição para casos que realmente não a têm muito superior ao trabalho citado. Ao compararmos a metodologia proposta com o trabalho de [Silva, 2016], percebemos uma proximidade nos valores dos resultados da acurácia, sensibilidade e especificidade. O trabalho de [Rocha, 2014] teve um melhor resultado em relação à acurácia e especificidade, mas a metodologia proposta teve um resultado melhor na sensibilidade, tornando-o melhor na capacidade em identificar corretamente a condição para casos que realmente tem. A curva ROC apresentada pela metodologia desenvolvida é maior do que os demais trabalhos, mostrando que a mesma consegue detectar um número maior de casos positivos.

Conclusão

Sistemas de Detecção e Diagnósticos Auxiliados por Computador (*Computer Aided Detection/Diagnosis*) vêm sendo usados e propostos como forma de auxílio aos profissionais de saúde, devido alguns casos os dados a ser analisados são de difícil avaliação por um especialista.

A utilização da rede neural CNN com arquitetura da rede *U-Net* para a extração das ROI's e a análise dessas imagens por meio dos Índices de Diversidade Filogenética, mostrou-se eficiente para a diferenciação entre a categoria maligna e benigna. A metodologia proposta apresentou uma taxa de acerto de 90,5%, uma sensibilidade de 92,6%, uma especificidade de 88,6% e uma curva ROC de 0,967. O dado mostrado pela

curva ROC mostra que a metodologia desenvolvida consegue detectar um alto número de casos que são positivos, ou seja, que possuem massas malignas.

A partir dos resultados obtidos, pode-se dizer que a utilização da rede neural CNN com arquitetura da rede *U-Net* para extração da ROI e a utilização de descritores baseados na textura, apresentam resultados eficazes na classificação em maligno e benigno para imagens de mamografias.

References

- Bochartt, T. B. et al. (2013). Análise de imagens termográficas para a classificação de alterações na mama. tese doutorado em computação.
- Breiman, L. (2001). "randomforest". machine learning, v.45, n.1, p.5-32.
- Goodfellow, I. et al. Deep learning. p. 154-224.
- Health, M., B. et al. (2000). The digital database for screening mamography. citeseer, proceedings of the 5th international workshop on digital mammography. p. 212-218.
- INCA (2017). Tipos de câncer: mama.
- Matos, C. E. F. et al. (2017). Diagnóstico de câncer de mama em imagens mamográficas através de características locais e invariantes.
- Neves, J. C. R. et al. (2011). Licenciatura em engenharia informática, departamento de engenharia informática.
- Oliveira, F. S. S. et al. (2013). Classificação de tecidos da mama em massa e não-massa usando índice de diversidade taxonômica e máquina de vetor de suporte. dissertação de mestrado.
- Resmini, R. et al. (2011). Análise de imagens térmicas da mama usando descritores de textura.
- Rocha, S. V. et al. (2014). Diferenciação de padrão de malignidade e benignidade de massas em mamografias utilizando características geométricas e máquina de vetor de suporte. dissertação de mestrado.
- Ronneberger, O., P.Fischer, and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer. (available on arXiv:1505.04597 [cs.CV]).
- Silva, T. F. d. B. S. et al. (2016). Diferenciação de padrão de malignidade e benignidade de massas em mamografias utilizando características geométricas e máquina de vetor de suporte. dissertação de mestrado.
- Webb, C. O. et al. (2000). Exploring the phylogenetic structure of acological communities: na example for rain forest trees. the american naturalist 156.2, p. 145-155.

Axiom Acquisition in STRIPS Planning Domains

Romulo O. Leite¹, Volmir E. Wilhelm¹

¹Programa de Pós-Graduação em Métodos Numéricos em Engenharia
Universidade Federal do Paraná (UFPR)
Caixa Postal 19.011 – 81.531-980 – Curitiba – PR – Brazil

{romulool,volmir}@ufpr.br

***Abstract.** Knowledge about domain axioms is useful in the modelling process. However, these axioms are not explicitly given from data and their formulation is not a trivial task. The present work is concerned about discovering domain axioms from a set of fully observed states. An experiment with three benchmark domains show the effectiveness of the proposed technique.*

1. Introduction

Modelling a STRIPS domain is a tough task, even in simple cases. This difficulty has attracted part of planning researchers community to focus on the modelling process, specially because most of planning techniques depends on well-formulated models. Such dependence represents a bottleneck for the development of planning technologies [Kambhampati 2007].

Some authors like [Yang et al. 2007] and [Zhuo et al. 2013] present methods to build models from data; particularly, they make use of a set of observations about the domain as a knowledge base, such as plan cases that are known to be correctly solved. In addition, [Ivankovic and Haslum 2015] show that knowledge about domain axioms is very useful as a support to the modelling process.

However, these axioms are not explicitly given by the set of observations; methods found in the literature require that such information must be provided. This feature leads to a controversy, because domain-specific knowledge is needed in order to apply a domain-independent method.

The present work is concerned specifically about the acquisition of domain axioms from the set of fully observed states. Using the information about these states, rules that must be valid are gathered by induction.

The next sections are organized as follows. Section 2 presents some definitions and briefly describes the problem of axiom acquisition in STRIPS domains. Section 3 presents the proposal of discovering domain axioms from fully observable states in details. Section 4 presents some results of the application of the proposed method in three benchmark domain models, besides of some discussion about those results. Section 5 concludes the present work.

2. Preliminaries and Definitions

This sections presents definitions required to the comprehension of the proposed method and describes the problem of discovering domain axioms using induction.

2.1. STRIPS Domains

Let L be a function-free first-order language with finitely many predicate symbols and finitely many constant symbols; a STRIPS planning domain on L is defined as a restricted state-transition system $\Sigma = (S, A, \gamma)$, such that S is a finite set of STRIPS states, A is a set of ground instances of STRIPS operators of a set O and γ is the transition function [Ghallab et al. 2004].

The set $O = \{o_1, \dots, o_k\}$ is such that each operator o_i represents a singular transition pattern. A STRIPS operator $o \in O$ is defined as a 3-tuple $(name(o), pre(o), eff(o))$, where $name(o)$ is the operator's name, $pre(o)$ is the set of preconditions of o and $eff(o)$ is the set of effects of o [Leite and Wilhelm 2015]. The operator's name, $name(o)$, is a syntactical expression of the form $n(?z_1, \dots, ?z_l)$ where n is a unique symbol and $(?z_1, \dots, ?z_l)$ is the l -tuple with the ordered arguments of the operator. Furthermore, $pre(o)$ and $eff(o)$ are both sets of ungrounded literals.

For the purpose of the present work, domain models are assumed to be typed, that is, every constant symbol represents an object to which is designated one type t_i of a set of types $T = \{t_1, \dots, t_n\}$. Consequently, every variable is associated to one type, such that a variable associated with type t_i can only assume values correspondent to objects of type t_i .

Let $P = \{p_1, \dots, p_m\}$ be the set of predicate symbols; every element $p_i \in P$ has the form $p(?w_1, \dots, ?w_r)$, where $(?w_1, \dots, ?w_r)$ is the r -tuple with the ordered argument variables of p . Eventually, r may be equal to zero. In addition, every variable of $(?w_1, \dots, ?w_r)$ has a designation to one type $t \in T$.

Let $V = \{v_1, \dots, v_n\}$, be a set such that every element $v_i = (v_{i1}, \dots, v_{iq_i})$ is a q_i -tuple with variables of type i . The number of variables of type i , q_i , is the maximum of the arguments of type i in the predicates of P . Furthermore, if p^* is a predicate with q_i arguments of type i , the order of the elements in v_i is the same as the order in which these elements figure in the list of arguments of p^* .

Let $s \in S$ be a STRIPS state; s is represented as a conjunction of atoms $\alpha_1 \wedge \dots \wedge \alpha_h$. Every atom is a literal of the form $\alpha_i = p_{\alpha_i}(w_1, \dots, w_r)$, where p_{α_i} is a predicate symbol and (w_1, \dots, w_r) is the ground r -tuple with the objects related by p_{α_i} .

2.2. Axioms and Domain Modelling

The task of building a model for a STRIPS domain basically consists on determining each operator of O , that is, the composition of its sets of preconditions and effects. Such task is mentioned by authors like [Weber and Bryce 2011] and [Zhuo et al. 2013] as a hard, time-consuming and error-prone task, what has motivated proposals to the use of computational methods in the modelling process. Many of these proposals are related to learning models from data; [Yang et al. 2007], [Gregory and Cresswell 2015] and [Zhuo 2015] are some examples.

Besides of learning models from plan traces, [Leite and Wilhelm 2015] also make use of domain axioms to guide the modelling process. According to [Ivankovic and Haslum 2015], the use of axioms provides more compact and natural formulations of the domain models.

An axiom is a rule $\pi \rightarrow \xi$, where π and ξ are ungrounded literals. In the context of planning domains, an axiom states that if π holds in some state, then ξ must also hold in that state. In the present work, we are particularly interested in two types of axioms: the ones that are valid whenever ξ is the consequent (which we call "type I" or "always-false" rules, because no ground instance of ξ is found in any state); and the ones where π is a positive literal and ξ is a negative literal, that is, rules that associate pairs of statements that cannot be valid at the same time (which we call "type II"). This last type of rule is specially useful to discard negative preconditions and derive delete effects from add effects in domain models.

Given a set of fully observed states in the domain Σ , the problem of axiom acquisition consists on finding a set of rules in the form $\pi \rightarrow \xi$ that must be valid in Σ . This problem may be solved by induction: if, for all the observed states, when a ground instance of a literal π and a ground instance of a literal ξ do not hold simultaneously, the rule that associate that pair of literals is induced in its ungrounded form. Furthermore, if a literal ξ is not found in any observed state, then its ungrounded form is classified as an always-false relation.

For example, consider Blocks World, a well-known benchmark domain [Zhuo et al. 2013] and let a be a block; if $ontable(a)$ holds in some state, $holding(a)$ will not hold for that state. This is because the literal $ontable(a)$ is equivalent to the statement "Block a in upon the table", whereas $holding(a)$ means "Block a is held by the crane" and, clearly, both statements cannot be true simultaneously. Hence, the axiom $ontable(?x) \rightarrow \neg(holding(?x))$ may be assumed to be valid. Consequently, the rule $holding(?x) \rightarrow \neg(ontable(?x))$ is derived by contraposition.

In addition, $on(a, a)$ will never hold in any state of Blocks World. This is because it is equivalent to the statement "Block a is placed on block a ", what is naturally incompatible with the semantic properties of the domain. Hence, $on(?x, ?x)$ is assumed to be always false and every rule of the type $\pi \rightarrow \neg(on(?x, ?x))$ is valid in Blocks World domain for any literal π .

3. Discovering Axioms by Induction

Basically, inducing an axiom consists on generalizing a rule from a particular observation. Once a rule is assumed to be valid, it will be discarded only if some latter observation brings a conflict that makes that rule no longer valid. The set of rules that remain valid over all the observations compose the set of axioms for a particular domain.

The induction method used in the present work is divided in two phases. Firstly, a search for always-false rules is made and, next, the set of rules with a positive antecedent and a negative consequent is built.

To induce type I rules, for every $p(?w_1, \dots ?w_r) \in P$, the permutation with repetition on the elements of $(?w_1, \dots ?w_r)$ is preformed and, for every particular permutation $?w$, a search in the set of observed states is made in order to find some occurrence of a ground instance of that permutation. If no occurrence is found, $p(?w)$ is added to always-false set. Once the permutation belongs to always-false set, rules of the form $\pi \rightarrow \neg(p(?w))$ are valid for any π .

The next step is to induce type II axioms. Let $\alpha = p_\alpha(w_1, \dots w_r)$ be an atom

in an observable state s and let T^* be the subset of T which contains the types of the objects in the set of arguments of α , that is, if every t_{w_i} is the type of the object w_i , then $T^* = \{t_{w_1}, \dots, t_{w_r}\} \cap T$. Define V^* as the subset of V such that each element in V^* is a tuple of V with variables associated to the types in T^* . Finally, define $V^*(\alpha)$ as a partial instance of V^* where each object in the set of arguments of α replaces one variable in crescent index order; if, for each type, the number of variables is greater than the number of objects, the exceeding variables remain ungrounded.

Each literal $\alpha \in s$ is taken as the antecedent of a batch of rule sketches. The consequent of each rule sketch is a literal formed by the permutation of the elements in the tuples of $V^*(\alpha)$. If no literal matches the form of the consequent in the observed state, then the ungrounded form of the rule is added to the set of axioms. Otherwise, if that rule have been previously added to the set of axioms, then it must be removed. Table 1 shows the pseudocode for axiom discovery algorithm.

Table 1. Pseudocode of axiom discovery algorithm

Input: a set S of observed STRIPS states, a set P of predicates, a set V of variables and a set T of types

```

01:  $R_I = \emptyset$ ;
02: for each  $p \in P$ :
03:     for each permutation  $p^*$  of arguments of  $p$ :
04:          $always\_false = \text{True}$ ;
05:         for each  $s \in S$ :
06:             if there is some literal in  $s$  which matches the form of  $p^*$ :
07:                  $always\_false = \text{False}$ ;
08:         if  $always\_false == \text{True}$ :
09:              $R_I.append(p^*)$ ;
10:  $R_{II} = \emptyset$ ;
11: for each  $s \in S$ :
12:     for each literal  $\alpha \in s$ :
13:         compute set  $T^*$  of types of arguments in  $\alpha$ ;
14:         compute set  $V^*$  with variables related to arguments in  $\alpha$ ;
15:         compute set  $V^*(\alpha)$  as a partial instance of  $V^*$  with objects in  $\alpha$ ;
16:         for each  $p \in P$ :
17:             for each permutation  $p^*$  of  $p$  with elements of  $V^*(\alpha)$ :
18:                  $sketch = (\alpha, p^*)$ ;
19:                  $\pi = ?\alpha$ ;
20:                  $\xi_{p^*} = ?p^*$ ;
21:                  $rule = [\pi \rightarrow \neg(\xi_{p^*})]$ ;
22:                 if there is no literal  $\alpha' \neq \alpha$  in  $s$  that matches the form of  $p^*$ :
23:                     if rule not in  $R_{II}$ :
24:                          $R_{II}.append(rule)$ ;
25:                     else:
26:                         if rule in  $R_{II}$ :
27:                              $R_{II}.remove(rule)$ .
```

Output: a set R_I of type I axioms and a set R_{II} of type II axioms

Figure 1 illustrates an example of an observed state in Blocks World. There are four blocks identified by A , B , C and D . Block C is on block A , what makes A not manipulable; blocks A , B and D are on the table and the crane is not holding any block.

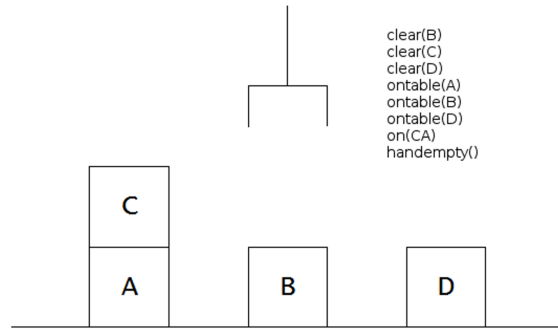


Figure 1. Example of an observed state in Blocks World with the corresponding STRIPS description

Let $P = \{clear(?x_1), ontable(?x_1), on(?x_1, ?x_2), holding(?x_1), handempty()\}$ be the set of predicates of Blocks World domain and assume that all objects are of type *block*. Consequently, $T = \{block\}$ is the set of types and $V = \{(?x_1, ?x_2)\}$. In this case, $T^* = T$ and $V^* = V$ are invariants.

Let s_0 be the depicted state. The atom $clear(B)$ is valid in s_0 ; so, for the first batch of rule sketches, assume $\alpha = clear(B)$ and $\pi = clear(?x_1)$. Consequently, $V^*(clear(B)) = \{(B, ?x_2)\}$.

Assume $p = ontable(?x_1)$; so, the permutations formed over $V^*(clear(B))$ are $p^*_1 = ontable(B)$ and $p^*_2 = ontable(?x_2)$ and the respective consequents are $\xi_{p^*_1} = ontable(?x_1)$ and $\xi_{p^*_2} = ontable(?x_2)$. For p^*_1 , it is verified that the rule sketch $clear(B) \rightarrow \neg(ontable(B))$ is not valid because block B is simultaneously clear and on the table. For p^*_2 , it is verified that the rule sketch $clear(B) \rightarrow \neg(ontable(?x_2))$ is also not valid because some other block (C , for example) is on the table while B is clear. Hence, the rules $clear(?x_1) \rightarrow \neg(ontable(?x_1))$ and $clear(?x_1) \rightarrow \neg(ontable(?x_2))$ are assigned to be false.

Now, assume $p = on(?x_1 ?x_2)$; so, the permutations formed over $V^*(clear(B))$ are $p^*_1 = on(B, B)$, $p^*_2 = on(B, ?x_2)$, $p^*_3 = on(?x_2, B)$ and $p^*_4 = on(?x_2, ?x_2)$ and the respective consequents are $\xi_{p^*_1} = on(?x_1, ?x_1)$, $\xi_{p^*_2} = on(?x_1, ?x_2)$, $\xi_{p^*_3} = on(?x_2, ?x_1)$ and $\xi_{p^*_4} = on(?x_2, ?x_2)$. For $\xi_{p^*_1}$ and $\xi_{p^*_4}$, it is verified that the consequent of the rule sketches $clear(B) \rightarrow \neg(on(B, B))$ and $clear(B) \rightarrow \neg(on(?x_2, ?x_2))$ have the form of one of the always-false relation and, hence, had already been assigned in type I induction phase. For $\xi_{p^*_2}$ and $\xi_{p^*_3}$, it is verified that the rule sketches $clear(B) \rightarrow \neg(on(B, ?x_2))$ and $clear(B) \rightarrow \neg(on(?x_2, B))$ are both valid, because there is no block either under or over B ; hence, the rules $clear(?x_1) \rightarrow \neg(on(?x_1, ?x_2))$ and $clear(?x_1) \rightarrow \neg(on(?x_2, ?x_1))$ are added to the axioms list. Note that $clear(?x_1) \rightarrow \neg(on(?x_1, ?x_2))$ is not an actual axiom of Blocks World; however, later in the analysis of s_0 , this assignment mistake will be detected, precisely when $\alpha = clear(C)$ and $p = on(?x_1 ?x_2)$, because block C is clear and is over block A .

4. Experiment

This section presents an experiment with three benchmark domains in order to test the axiom acquisition algorithm described in Section 3. Blocks World and Logistics [Bacchus 2001] are planning domains used in Artificial Intelligence Planning and Scheduling (AIPS) competition in the year 2000; Dock Worker Robot (DWR) domain is presented by [Ghallab et al. 2004] and used for didactic and illustrative purposes.

For each domain, the input is composed of a set of 100 observed states and a set of predicates with respective arguments (variables and types).

4.1. Blocks World

The environment of Blocks World is a table where blocks are laid upon. There is a crane used to form piles with the blocks; these piles has no limit of height. Also, there is no limit for the number of blocks and the table is assumed to be large enough to support all of them. Nonetheless, the number of blocks is assumed to be constant. All blocks are identified with a label and have the same dimensions.

Table 2 presents the axioms discovered for the Blocks World domain with respective interpretations. All the 19 rules found are actual axioms for Blocks World.

Table 2. Axioms discovered for Blocks World domain

Axiom	Interpretation
$\pi \rightarrow \neg(\text{on}(\text{?}x_1, \text{?}x_1))$	a block cannot be over itself
$\pi \rightarrow \neg(\text{on}(\text{?}x_2, \text{?}x_2))$	a block cannot be over itself
$\text{clear}(\text{?}x_1) \rightarrow \neg(\text{on}(\text{?}x_2, \text{?}x_1))$	if a block is clear, then no other block is upon it
$\text{clear}(\text{?}x_1) \rightarrow \neg(\text{holding}(\text{?}x_1))$	if a block is clear, then it is not held by the crane
$\text{ontable}(\text{?}x_1) \rightarrow \neg(\text{on}(\text{?}x_1, \text{?}x_2))$	if a block is on the table, then it is not on another block
$\text{ontable}(\text{?}x_1) \rightarrow \neg(\text{holding}(\text{?}x_1))$	if a block is on the table, then it is not held by the crane
$\text{on}(\text{?}x_1, \text{?}x_2) \rightarrow \neg(\text{on}(\text{?}x_2, \text{?}x_1))$	”on” is not a reciprocal relation
$\text{on}(\text{?}x_1, \text{?}x_2) \rightarrow \neg(\text{ontable}(\text{?}x_1))$	if a block is on another, then it is not on the table
$\text{on}(\text{?}x_1, \text{?}x_2) \rightarrow \neg(\text{clear}(\text{?}x_2))$	if a block is under another, then it is not clear
$\text{on}(\text{?}x_1, \text{?}x_2) \rightarrow \neg(\text{holding}(\text{?}x_1))$	if a block is on another, then it is held by the crane
$\text{on}(\text{?}x_1, \text{?}x_2) \rightarrow \neg(\text{holding}(\text{?}x_2))$	if a block is under another, then it is held by the crane
$\text{handempty}() \rightarrow \neg(\text{holding}(\text{?}x_1))$	if the crane is empty, then it is not holding a block
$\text{handempty}() \rightarrow \neg(\text{holding}(\text{?}x_2))$	if the crane is empty, then it is not holding a block
$\text{holding}(\text{?}x_1) \rightarrow \neg(\text{on}(\text{?}x_1, \text{?}x_2))$	if a block is held by the crane, then it is not on another block
$\text{holding}(\text{?}x_1) \rightarrow \neg(\text{on}(\text{?}x_2, \text{?}x_1))$	if a block is held by the crane, then it is not under another block
$\text{holding}(\text{?}x_1) \rightarrow \neg(\text{ontable}(\text{?}x_1))$	if a block is held by the crane, then it is not on the table
$\text{holding}(\text{?}x_1) \rightarrow \neg(\text{clear}(\text{?}x_1))$	if a block is held by the crane, then it is not clear
$\text{holding}(\text{?}x_1) \rightarrow \neg(\text{handempty}())$	if a block is held by the crane, then the crane is not empty
$\text{holding}(\text{?}x_1) \rightarrow \neg(\text{holding}(\text{?}x_2))$	the crane can hold only one block at a time

4.2. Logistics

The Logistics domain is a simplified model for the operations of a logistics company. There are a certain number of cities and a certain number of locations in each city. Each location contains a warehouse where the company stock packages to be delivered. There is one truck per city; each truck is used to carry packages between locations inside a city. At least one location per city have an airport where planes can land. Planes are used to carry packages between cities and there is no limit for the number of planes.

Table 3 presents the axioms discovered for the Logistics domain with respective interpretations. All the 4 rules found are actual axioms for Logistics.

Table 3. Axioms discovered for Logistics domain

Axiom	Interpretation
$stored(?p, ?l_1) \rightarrow \neg(in(?p, ?t))$	a package stored in a location is not loaded in a truck
$stored(?p, ?l_1) \rightarrow \neg(aboard(?p, ?a))$	a package stored in a location is not aboard a plane
$aboard(?p, ?a) \rightarrow \neg(stored(?p, ?l_1))$	a package aboard a plane is not stored in a location
$in(?p, ?t) \rightarrow \neg(stored(?p, ?l_1))$	a package loaded in a truck is not stored in a location

4.3. DWR

The DWR domain is a prototype of a port where containers are carried by robots. These robots move between adjacent locations and have load capacity for only one container. There must be at maximum one robot per location at a time. Each location has a crane used to manipulate containers. Each location has also a certain number of pallet which serve as support for container piles.

Table 4 presents the axioms discovered for the DWR domain with respective interpretations. All the 27 rules found are actual axioms for DWR.

Table 4. Axioms discovered for DWR domain

Axiom	Interpretation
$\pi \rightarrow \neg(adjacent(?l_1, ?l_1))$	a location is not adjacent to itself
$\pi \rightarrow \neg(adjacent(?l_2, ?l_2))$	a location is not adjacent to itself
$\pi \rightarrow \neg(on(?c_1, ?c_1))$	a container cannot be on itself
$\pi \rightarrow \neg(on(?c_2, ?c_2))$	a container cannot be on itself
$attached(?p, ?l_1) \rightarrow \neg(attached(?p, ?l_2))$	a pile is attached to only one location
$belong(?k, ?l_1) \rightarrow \neg(belong(?k, ?l_2))$	a crane belongs to only one location
$at(?r, ?l_1) \rightarrow \neg(at(?r, ?l_2))$	a robot can be in only one location
$at(?r, ?l_1) \rightarrow \neg(free(?l_1))$	if a robot is at a location, then the location is not free
$free(?l_1) \rightarrow \neg(at(?r, ?l_1))$	if a location is free, then there is no robot at it
$unloaded(?r) \rightarrow \neg(loaded(?r, ?c_1))$	if a robot is unloaded, then it carries no container
$unloaded(?r) \rightarrow \neg(loaded(?r, ?c_2))$	if a robot is unloaded, then it carries no container
$holding(?k, ?c_1) \rightarrow \neg(loaded(?r, ?c_1))$	if a container is held, then it is not loaded in a robot
$holding(?k, ?c_1) \rightarrow \neg(holding(?k, ?c_2))$	a crane can hold only one container
$holding(?k, ?c_1) \rightarrow \neg(empty(?k))$	if a crane holds a container, then it is not empty
$holding(?k, ?c_1) \rightarrow \neg(in(?c_1, ?p))$	if a container is held by a crane, then it is not in any pile
$holding(?k, ?c_1) \rightarrow \neg(top(?c_1, ?p))$	a container held by a crane is not at the top of any pile
$empty(?k) \rightarrow \neg(holding(?k, ?c_1))$	if a crane is empty, then it is not holding any container
$empty(?k) \rightarrow \neg(holding(?k, ?c_2))$	if a crane is empty, then it is not holding any container
$in(?c_1, ?p) \rightarrow \neg(holding(?k, ?c_1))$	if a container is in a pile, then it is not held by a crane
$top(?c_1, ?p) \rightarrow \neg(loaded(?r, ?c_1))$	a container at the top of a pile is not loaded in a robot
$top(?c_1, ?p) \rightarrow \neg(holding(?k, ?c_1))$	if container is top of a pile, then it is not held by a crane
$top(?c_1, ?p) \rightarrow \neg(top(?c_2, ?p))$	a pile has only one top container
$on(?c_1, ?c_2) \rightarrow \neg(on(?c_2, ?c_1))$	"on" is not a reciprocal relation
$loaded(?r, ?c_1) \rightarrow \neg(loaded(?r, ?c_2))$	a robot can carry only one container
$loaded(?r, ?c_1) \rightarrow \neg(unloaded(?r))$	if a robot carries a container, then it is not unloaded
$loaded(?r, ?c_1) \rightarrow \neg(holding(?k, ?c_1))$	a container loaded in a robot is not being held by a crane
$loaded(?r, ?c_1) \rightarrow \neg(top(?c_1, ?p))$	a container loaded in a robot is not at the top of a pile

4.4. Discussion

The results of the experiment reported in the previous section illustrate the effectiveness of the proposed technique to discover domain axioms from a set of observed states. It is remarkable that these results rely on a sufficient amount and broad variability of the data

used as input; a small set of states with high degree of similarity between observations constitutes the worst scenario for the application of the algorithm.

The list of type I and type II axioms for Blocks World and DWR are complete, that is, all the rules of those types were found. However, there are more axioms in Logistics than those reported. Rules related to locations were not discovered; for example, $landed(?a, ?l_1) \rightarrow \neg(landed(?a, ?l_2))$ states that a plane cannot be at two locations at the same time and it is clearly a valid rule that was not reported. This lack of completeness is due to a particularity of Logistics domain: there is no predicate that relate two different locations and, therefore, there is only one variable of type "location". This particular result indicates a limitation of the proposed technique.

Due to the toughness of the task of discovering axioms, the algorithm has a relatively high complexity. In both Type I and II axiom discovery phases, the number of operations is dominated by r , that is, the lengths of the sets of arguments of the predicates. As a permutation with repetition in those arguments is performed, the computational effort grows exponentially in terms of r . However, a predicate that relates lots of objects is not of practical use; typically, a predicate has 3 arguments at maximum. Furthermore, a complex relation generally can be split into simpler ones. So, in the worst case, the algorithm runs in cubic time.

Each pair of literals in an axiom is a generalization of ground literals that must not figure simultaneously in a state of the respective domain. The knowledge about these axioms may be applied, for example, as a criterion for the identification of valid states or as a set of constraints in a random states simulator. Furthermore, if domain axioms are known, given a planning problem with a non-empty goal set, it is possible to determine whether such problem is feasible before the application of some problem-solving technique by verifying if some pair of goal literals match some of the axioms, resulting in time and computational efforts saving.

5. Conclusions

The objective of the present work is to present a method to extract domain axioms from a set of fully observable states. Such method works by generalizing rules from particular cases and testing these rules against other observations. An application of the algorithm to three benchmark domains evidence its effectiveness in discovering valid rules, despite of a limitation in terms of completeness. Future work will be dedicated to overcome such limitation and submit the algorithm to tests in domain models with higher degrees of complexity.

Future work will also be concerned in aggregate this technique to a new tool for STRIPS domains modelling.

References

- Bacchus, F. (2001). Aips'00 planning competition: The fifth international conference on artificial intelligence planning and scheduling systems. *AI Magazine*, 22(3):47–56.
- Ghallab, M., Nau, D., and Traverso, P. (2004). *Automated Planning: Theory and Practice*. Morgan Kaufmann.

- Gregory, P. and Cresswell, S. (2015). Domain model acquisition in the presence of static relations in the lop system. In *Proceedings of the XXV International Conference on Automated Planning and Scheduling*, pages 97–105. AAAI.
- Ivankovic, F. and Haslum, P. (2015). Optimal planning with axioms. In *Proceedings of the XXIV International Joint Conference on Artificial Intelligence*, pages 1580–1586.
- Kambhampati, S. (2007). Model-lite planning for the web age masses: The challenges of planning with incomplete and evolving domain models. In *Proceedings of the XXII AAAI Conference on Artificial Intelligence*, pages 1601–1605. AAAI.
- Leite, R. and Wilhelm, V. (2015). An action model learning method for planning in incomplete strips domains. In *Proceedings of the XII Encontro Nacional de Inteligência Artificial e Computacional*, pages 14–19. SBC.
- Weber, C. and Bryce, D. (2011). Planning and acting in incomplete domains. In *Proceedings of the XXI International Conference on Automated Planning and Scheduling*, pages 274–281. AAAI.
- Yang, Q., Wu, K., and Jiang, Y. (2007). Learning action models from plan examples using weighted max-sat. *Artificial intelligence Journal*, 171:107–143.
- Zhuo, H., Nguyen, T., and Kambhampati, S. (2013). Refining incomplete planning domain models through plan traces. In *Proceedings of the XXIII International Joint Conference on Artificial Intelligence*, pages 2451–2457.
- Zhuo, H. H. (2015). Crowdsourced action-model acquisition for planning. In *Proceedings of the XXIX AAAI Conference on Artificial Intelligence*, pages 3439–3445. AAAI.

DependentIE: An Open Information Extraction system on Portuguese by a Dependence Analysis

Leandro Souza de Oliveira, Rafael Glauber, Daniela Barreiro Claro

¹Formalisms and Semantic Applications Research Group (FORMAS)

LASiD/DCC/IME – Federal University of Bahia (UFBA)

Av. Adhemar de Barros, s/n, Ondina, Salvador, Bahia, Brazil

leo.053993@gmail.com, rglauber@dcc.ufba.br, dclaro@ufba.br

Abstract. *The amount of textual data available on the Web grows every day. For humans, it is hard to manual extract useful information from such quantity of data. Traditional approaches extract information in a limited manner within a set of pre-determined features. Open information extraction (Open IE) extracts facts from plain texts without establishing the type of a relationship previously. In this work, we describe our DependentIE, an Open IE system for texts written in Portuguese that uses dependency analysis to identify clauses in a sentence. Our approach differs from the other methods such as ClausIE and ArgOE because we do not use handcrafted rules to extract the facts. Our results confirm that our approach is more precise than the current state of the art for Portuguese texts.*

1. Introduction

Nowadays the Web publishes a lot of data as texts written in a natural language format. The number of those texts are growing every day. Nevertheless, it is hard for humans to manual extract useful information from such a large data repository. Information extraction (IE) is a research area that automates the extraction of facts from textual documents [de Abreu et al. 2013]. According to Fader *et al.* [Fader et al. 2011] and Xavier *et al.* [Xavier et al. 2015], traditional IE is based on training an extractor with a target relationships previously defined. The main drawback of traditional IE approaches is their low coverage and their overfitting when applicable in a particular domain. Tackling with different domains a human intervention might be required. To overcome this problem, Open Information Extraction (Open IE) has emerged to extract facts without determining the set of relationships previously. Open IE extracts facts from sentences in triple forms, such as:

$$t = (arg1, rel, arg2) \quad (1)$$

where *arg1* and *arg2* are nominal phrases in a sentence and *rel* establishes a relationship between *arg1* and *arg2* through a verbal phrase. Estimates¹ show that 155 million of users on the Internet have Portuguese as their mother language. Considering the websites content, only 2.5% are written in Portuguese². Holding the importance of all other languages over the World, the first works in Open IE have taken their attentions to texts

¹<http://www.internetworldstats.com/stats7.htm>

²https://w3techs.com/technologies/overview/content_language/all

written in English. To the best of our knowledge, the Open IE work which presents the best results for English texts is called ClausIE [Del Corro and Gemulla 2013]. ClausIE is based on a set of manual rules and a dependency parser (DP) [Rodríguez et al. 2016]. For Portuguese languages, this kind of approach is carried by systems such as DepOE [Gamallo et al. 2012] and ArgOE [Gamallo and Garcia 2015]. Those systems define generic manual rules for different languages (multilingual systems). We do believe that compatible rules for different languages impose some limitations due to not dealing with particular aspects of each language. Different from them, our method extracts facts by a depth-first search (DFS) approach to identify the arguments. We can summarize our main contributions:

- We have created an Open IE system based on DP for Portuguese.
- We propose a new way of identifying relationship arguments using depth-first search.
- We create models for *tokenizer* tasks, *POS tagger* and *dependency parser* based on the *Universal Dependencies version 1.4* to Portuguese³.
- We have created two datasets with sentences written in Portuguese and from different domains to evaluate our approach.

The remainder of this work is organized as follows: Section 2 presents our approach called DependenteIE. Section 3 describes our experimental methodology and the materials used in our evaluation. In Section 4 we explain our results and discussed some insights in Section 5. Section 6 discusses the set of our closest works. Finally, Section 7 concludes and presents some envisioning work.

2. Our DependenteIE Approach

DependenteIE is an Open IE system for texts written in Portuguese language. As well as ArgOE [Gamallo and Garcia 2015] and ClausIE [Del Corro and Gemulla 2013] we use a Dependence Parser (DP) to identify clauses⁴ (useful parts of a sentence). In this work, a clause is one of the following parts of a sentence: subject (S), direct and indirect objects (O), verb (V), adverb (A), complement (C) and modifier (M). Our method extracts facts using clauses based on the standard SV (Subject - Verb). The arguments are detected through a deep-search in the sentence dependency.

The execution of our DependenteIE needs some preprocessing tasks, such as a Tokenizer, a Part-Of-Speech Tagger and a Dependence Parser for Portuguese (Figure 1). Those three analyzers mark each sentence.

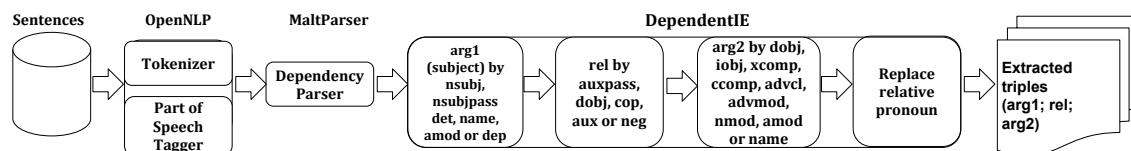


Figure 1. Overview of the process made by our DependenteIE.

³<http://universaldependencies.org/>

⁴The clauses consist of a subject and a verb and their constituents, such as objects (direct and indirect), adverbs and others.

Figure 2 depicts an example of a preprocessed output sentence. For instance, consider the sentence *A espada, que era uma peça de decoração, foi apreendida para ser periciada em 1967.* (The sword, which was a decorative piece, was seized for inspection in 1967.). For each token, our approach finds all dependents. The other subject-dependent token (*nsubjpass*) “espada” (sword) is defined as an article (*det*) “A” (the) and a noun (*acl:relcl*) “peça” (piece).

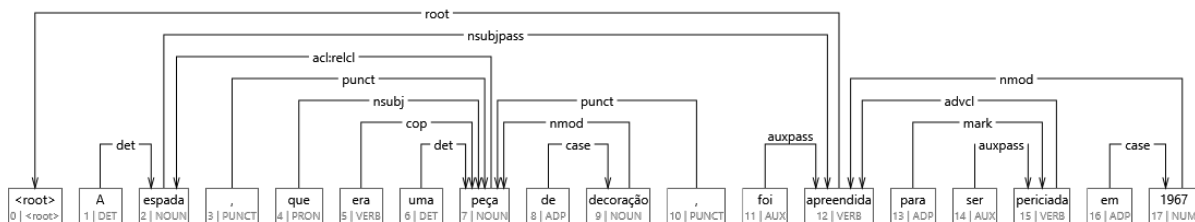


Figure 2. Example of a sentence with tokens and annotators of the POS tagger and the DP. Both Tokenizer and POS tagger were undertaken by Apache OpenNLP and the DP by MaltParser.

Taking our *DependentIE* (Figure 1), a sentence might contain a subject (*arg1*), a verbal phrase (*rel*) (*SV*) and one or more arguments (*arg2*). The identification of *arg1* starts locating the token identified by *nsubj* or *nsubjpass* in the dependency tree. In the next step, the following dependents are found: *det*, *name*, *amod*, *nmod* or *dep*. Considering our sentence “A espada” (The sword) and “which”. The identification of *rel* is performed by the subject HEAD concatenated with the dependents such as: *auxpass*, *dobj*, *cop*, *aux* or *neg*. In the end, we identify the *arg2* starting again from the head of the subject following by different paths such as:

- *SV + path + O* by *dobj*, *iobj*.
- *SV + path + C* by *xcomp* and *ccomp*.
- *SV + path + A* by *advcl* and *advmod*.
- *SV + path + PROPER_NAMES* by *name*.
- *SV + path + M* by *nmod* and *amod*.

Applying our *DependentIE* to our example we extract: $t_1 =$ (The sword/a espada, was seized/foi apreendida, for inspection/para ser periciada), $t_2 =$ (The sword/a espada, was seized/foi apreendida, in 1967/em 1967) and $t_3 =$ (which/que, was a piece/foi uma peça, decoration/de decoração). However, the t_3 triple does not represent a coherent information. To solve this issue, we replace the relative pronoun “que” (which) by the subject “A espada” (The Sword). This is done by identifying the dependency *acl:relcl* between the head of “que” and the subject “A espada”. When this type of fact is identified, our method always replaces the head with the subject as the example described above to extract facts more informative. In Table 1 we present all the facts extracted by our method from our sample sentence.

3. Experimental Setup

The performance of *DependentIE* was measured against two other Open IE systems on Portuguese: the *ArgOE* [Gamallo and Garcia 2015] and the proposal presented by Sena *et al.* (2017) in [Sena *et al.* 2017] (We call it *InferReVerbPT*). Each system received as

Table 1. Extracted triples

	arg1	rel	arg2
	A espada, que era uma peça de decoração, foi apreendida para ser periciada em 1967 / The sword, which was a decorative piece, was seized for inspection in 1967.		
Triple 1	A espada / The sword	foi apreendida / was seized	para ser periciada / for inspection
Triple 2	A espada / The sword	foi apreendida / was seized	em 1967 / in 1967
Triple 3	A espada / The sword	era uma peça / was a piece	de decoração / decoration

input two datasets composed by sentences. The first dataset was built using the Corpus of Electronic Texts Extracts NILCS/*Folha de São Paulo* newspaper (CETENFolha)⁵ version 2008. The second dataset was built from the digital encyclopedia *Wikipedia*⁶. Both datasets were composed of 200 sentences which were retrieved randomly. Those datasets were called CETEN200 and WIKI200 respectively. The selection of the sentences follows this set of criteria:

- A sentence must contain three or more tokens.
- A sentence should end with the dot (.) character.
- A sentence must be initiated with a capital letter.
- A sentence must have at least a noun or a pronoun.
- A sentence should not be initiated by a conjunction.

To evaluate the extraction by our DependentIE, two experts labeled the facts extracted from all three systems mentioned earlier. We consider two aspects of each extracted fact: (i) coherent or incoherent and (ii) minimal or not. A fact is considered coherent (i) if it keeps the same meaning as in the original sentence. A coherent fact may contain another fact in its arguments[Bast and Haussmann 2013]. In the sentence *Dener morreu no dia 19 de abril num acidente de carro, no Rio*. (Dener died on April 19 in a car accident in Rio.) a system can extract (Dener; morreu no; dia 19 de abril num acidente de carro no Rio) (Dener; died; on 19 April in a car accident in Rio). The fact is coherent, but we can still extract new coherent facts such as: (Dener; morreu; no dia 19 de Abril) / (Dener; died; on 19 April), (Dener; morreu; num acidente de carro) / (Dener; died; in a car accident) e (Dener; morreu; no Rio) / (Dener; died; in Rio). A fact which cannot be decomposed into new facts is considered as a Minimum fact (ii). Minimum facts can be more precise in Question Answering (QA) systems and either in the construction of ontologies [Bast and Haussmann 2013]. We measure the precision of our system by the ratio between all the extracted facts and their coherence. We use this ratio to measure the overall precision in *precision-c* Eq. 2 and by minimality *precision-m* Eq. 3. Considering the metric "recall", in OpenIE area, it is a difficult metric to calculate. Due to OpenIE

⁵<http://www.linguateca.pt/cetenfolha/>

⁶<https://pt.wikipedia.org/>

methods perform their extractions in an open domain, it is hard to estimate all the false negatives of a system to calculate the "recall". Only one sentence may have other combinations and interpretations of facts that can result to generate new facts. It depends on the human interpretation/analysis of each sentence. Thus, in this work, we did not calculate the "recall" measurement because of its imprecise value in OpenIE domain.

$$precision - c = \frac{\#(coherent_triples)}{\#(extracted_triples)} \quad (2)$$

$$precision - m = \frac{\#(minimum_triples)}{\#(coherent_triples)} \quad (3)$$

An extracted fact is considered coherent or minimal if both evaluators agreed with each other. The agreement of the experts was verified by Cohen's kappa coefficient [Carletta 1996]. In Table 2 we present the degree of agreement between those two experts.

Table 2. Analysis of cohen's kappa in both datasets for coherence and minimality

	CETEN200	WIKI200
Kappa (coherent)	0,807	0,846
Kappa (minimality)	0,778	0,860

Both experts had almost perfect agreement in evaluating the coherence in both datasets. In the evaluation of minimality, both experts had a full agreement in CETEN200 and almost a complete agreement in WIKI200.

3.1. Materials

The DP used by DependentIE was *MaltParser* [Nivre et al. 2006] with the library *LIB-SVM* [Chang and Lin 2011]. To the best of our knowledge, there is no model for DP in Brazilian Portuguese language pattern *Universal Dependencies* version 1.4⁷ (UDv1.4). We train the DP using the *Bosque* treebank in the CoNLL format available in UDv1.4. The input of a DP is a separate sentence into tokens within a POS tagger annotation. For both tasks, we use the Apache OpenNLP⁸. As we did not find a model for both tools based on UDv1.4 for Portuguese, we trained our model for tokenizer with a subset of 9047 sentences randomly chosen from the treebank *Bosque* following the annotation presented in the Table 3. Our tokenizer model presented a high precision as shown in Table 4.

To train the POS tagger, we also used the *Bosque* treebank. All sentences were pre-processed to stay in the format shown in Table 5. We used 10-fold cross-validation. Our POS Tagger model presented 95.84% of accuracy.

The model created for DP was evaluated using the following metrics: Labeled attachment score (LAS), unlabeled attachment score (UAS) and label accuracy (LA). LAS checks the accuracy of both the relation and the syntactic analysis. The UAS checks the accuracy of tokens' relationships *tokens*. The LA checks the accuracy of the syntactic analysis of each *token*.

⁷http://universaldependencies.org/#pt_br

⁸<https://opennlp.apache.org/>

Table 3. Pre-processed sentences to train the Tokenizer

Sentences	Sentences Pre-processed to be trained
A espada, que era uma peça de decoração da casa, foi apreendida para ser periciada / The sword, which was a decorative piece of the house, was seized for inspection	A espada<SPLIT>, que era uma peça de decoração da casa<SPLIT>, foi apreendida para ser periciada / The sword<SPLIT>, which was a decorative piece of the house<SPLIT>, was seized for inspection
Morena (Nanda Costa) e Wanda (Totia Meirelles) declararam guerra uma à outra. / Morena (Nanda Costa) and Wanda (Totia Meirelles) declared war on each other.	Morena <SPLIT>(Nanda Costa <SPLIT>) e Wanda <SPLIT>(Totia Meirelles<SPLIT>) declararam guerra uma à outra<SPLIT>. / Morena <SPLIT>(Nanda Costa<SPLIT>) and Wanda <SPLIT>(Totia Meirelles<SPLIT>) declared war on each other<SPLIT>.

Table 4. Tokenizer results

Precision	Recall	F1-measure
99,99%	87,63%	93,40%

Table 5. Pre-processed sentences to train the POS tagger.

Sentences	Sentences Pre-processed to be trained
A espada, que era uma peça de decoração da casa, foi apreendida para ser periciada / The sword, which was a decorative piece of the house, was seized for inspection	A_DET espada_NOUN ,_PUNCT que_PRON era_VERB uma_DET peça_NOUN de_ADP decoração_NOUN da_ADP casa_NOUN ,_PUNCT foi_AUX apreendida_VERB para_ADP ser_AUX periciada_VERB / The_DET sword_NOUN,_PUNCT which_PRON was_VERB a_DET decorative_NOUN piece_NOUN of_ADP the_DET house_NOUN,_PUNCT was_AUX seized_VERB for_ADP inspection_VERB

Table 6. Dependency parser results

LAS	UAS	LA
81,7%	84,3%	91,4%

4. Results

We compared the *DependentIE* against *ArgOE* and *InferReVerbPT* in quantity of facts extracted and precision. We organized our results presenting initially the numbers obtained in CETEN200 and then in WIKI200. The amount of facts extracted in CETEN200 is shown in Figure 3(a) and the precision in the Figure 3(b). Our proposal extracted 359 facts of which 188 are coherent. We obtained 52.4% by *precision-c* and 69.7% by

precision-m. With more coherent facts extracted and a higher precision, we have the highest gain among all three systems compared in CETEN200. In a direct comparison with *ArgOE* that also uses DP, we had 33.5% of more coherent facts.

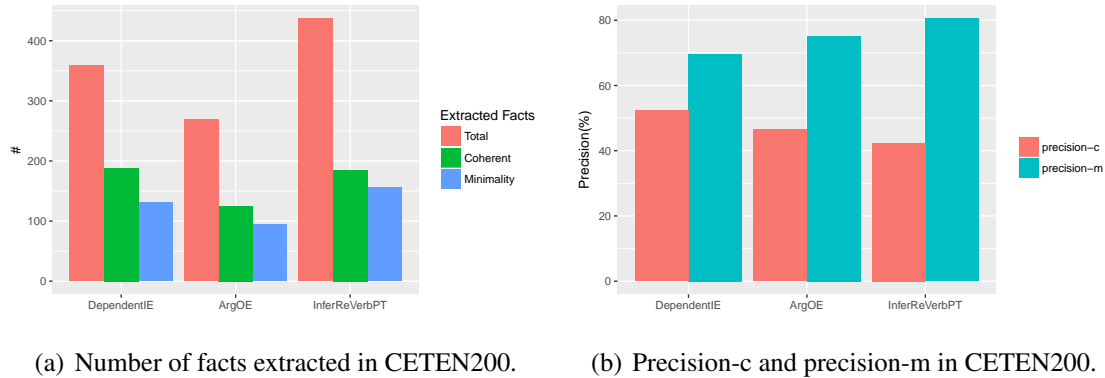


Figure 3. Result in CETEN200 dataset.

The results about the facts extracted in WIKI200 are presented in Figures 4(a) and 4(b). In WIKI200 the *DependentIE* was able to extract 279 triples of which 160 were coherent. Our method obtained 57.34% in *precision-c* and 85% of the coherent extracted facts were minimal. In WIKI200 the *ArgOE* performed better for *precision-m*. However, *DependentIE* continues to present the greatest gain among the approaches considering the number of coherent facts and *precision-c*. *InferReVerbPT* has a small advantage on the number of coherent facts. However, it has a low gain approach that is evident in WIKI200. In WIKI200 dataset our method can maintain a good proportion between the number of coherent facts and their minimality. These characteristics give our proposal an alternative application in Open IE for Portuguese texts in both scenarios.

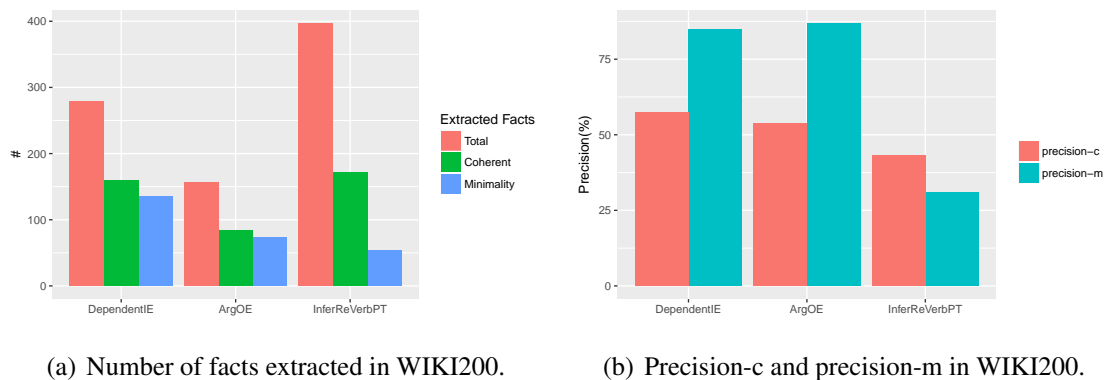


Figure 4. Result in WIKI200 dataset.

5. Discussion

There was a subset of sentences for which no extractions were obtained. In CETEN200 *InferReVerbPT* left to extract facts from 7 sentences and *ArgOE* left 85 sentences without extractions. Our *DependentIE* did not get extractions in 42 sentences. However, our approach presented the best proportion of coherent facts and *precision-c*. In WIKI200 the number of sentences without extraction increased for our *DependentIE* up to 66 sentences.

Even so, our proposal presents the highest gain among the evaluated systems. In most cases, the DP did not correctly label the sentence preventing the extraction of some facts. Another important observation was the behavior of our method for *precision-m* in both datasets. We believe that their writing style influenced on our experiments. CETEN200 is formed of journalistic texts, and the WIKI200 dataset is formed with encyclopedia texts thus confirming the different styles. For a better understanding of this kind of problem, we consider observing some examples of sentences from both datasets. On many occasions, the systems make different extractions. Each approach has different characteristics, and this reflects in the results. For example, in Table 7 the *DependentIE* had better performance by extracting 2 coherent and minimal triples. This happens because our proposal does a search in the dependency tree and we can recover more arguments than the other systems. In case of *ArgOE*, there was no extraction. We believe that this fact is justified by some error in DP as the researchers themselves emphasize. The *InferReVerbPT* made only a coherent extraction, but that it is not minimal. Methods that use shallow parsers have some difficulties retrieving arguments away from the verbal phrase.

Table 7. Example where *DependentIE* is better than the other methods.

Sentence pt-br/en: Esse asteroide foi descoberto em 24 de Maio de 1931 por Cyril Jackson. / This asteroid was discovered on 24, May in 1931 by Cyril Jackson.				
System	Triples (pt-br / en)	Coherent	Minimality	
DependentIE	(Esse asteroide; foi descoberto; em 24 de Maio de 1931) / (This asteroid; was discovered; on May 24 in 1931)	yes	yes	
	(Esse asteróide; foi descoberto; por Cyril Jackson) / (This asteroid; was discovered; by Cyril Jackson)	yes	yes	
ArgOE	-	-	-	
InferReVerbPT	(Esse asteroide; foi descoberto; em 24 de Maio de 1931 por Cyril Jackson) / (This asteroid; was discovered; on 24, May in 1931 by Cyril Jackson)	yes	no	

We present another example in the Table 8. The method *DependentIE* did not extract facts because the DP did not find the subject of the sentence. Meanwhile, *ArgOE* had a better performance by making both two coherent and minimal extractions. The *InferReVerbPT* made two incoherent extractions because it was wrong about the definition of the arguments in both cases.

In the example on Table 9, both *DependentIE* and *ArgOE* did not perform extractions. Once again the DP did not find a subject. However, the *InferReVerbPT* made a coherent and minimal extraction. The low precision of more sophisticated language tools for Portuguese is still a barrier to Open IE. While analyzers such as POS tagger and Chunker present a precision close to 100%⁹ the *MaltParser* present a precision near to 80% for the DP task. Thus, we believe that more effort is still needed to create efficient linguistic tools beyond the English language.

⁹InferReVerbPT uses the CoGrOO <http://ccsl.ime.usp.br/cogroo/>

Table 8. Example where ArgOE is better than the other methods.

Sentence pt-br/en: O primeiro serviço do MD-11 nos Estados Unidos ocorreu ainda em 1990 e foi,inaugurado pela Delta Airlines. / The first service of MD-11 in the United States occurred in 1990 and was inaugurated by Delta Airlines.

System	Triples (pt-br/en)	Coherent	Minimality
DependentIE	-	-	-
ArgOE	(O primeiro serviço do MD-11; ocorreu nos; Estados Unidos) / (The first service of MD-11; occurred in; United States)	yes	yes
	(O primeiro serviço do MD-11; ocorreu em; 1990) / (The first service of MD-11; occurred in; 1990)	yes	yes
InferReVerbPT	(O primeiro serviço do MD-11 nos Estados Unidos; ocorreu ainda em; 1990 foi inaugurado pela Delta Airlines) / (The first service of MD-11; occurred in; 1990 was inaugurated by Delta Airlines)	no	-
	(O primeiro serviço do MD-11 nos Estados Unidos ocorreu ainda em 1990; foi inaugurado pela; Delta Airlines) / (The first service of MD-11 occurred in 1990; was inaugurated; by Delta Airlines)	no	-

Table 9. Example where InferReVerbPT is better than the other methods.

Sentence pt-br/en: Põlde é localizado no município de Abja vald e Viljandimaa, no sul do país, e 150 km,ao sul da capital Tallinn. / Põlde is located in the municipality of Abja vald and Viljandimaa in the south of the country, and 150 km south of the capital Tallinn.

System	Triples (pt-br/en)	Coherent	Minimality
DependentIE	-	-	-
ArgOE	-	-	-
InferReVerbPT	(Põlde; é localizado no; município de Abja vald) / (Põlde; is located in the; municipality of Abja vald)	yes	yes

Overall, from a preliminary analysis, it was verified that in many cases the DependentIE made incoherent extractions due to the Dependency Parser produces a dependency tree with bad grammar structures. In the sentence "Para o Bank of America a opção foi a de não converter nenhum tostão." the DependentIE returned the subject "a de não" because the DP labeled the article "a" as "nsubj". As a consequence, DependentIE generates an incoherent triple (a de não; opção nenhum tostão; Para o Bank of America). We believe that the improvement of NLP tools can improve the quality of our DependentIE, allowing it to do more coherent extractions with both high precision-m and precision-c. Another problem detected concerns the Portuguese language. The extractions of triples are made based on a subject, but in Portuguese, there are implicit subjects, which does not exist in

English language. For example, in the sentence "faremos o trabalho" DependentIE does not extract any triples. This problem could be solved by using some tool to verify that there is an implicit subject (the pronoun "nós"), allowing the method to extract the triple (nós; faremos; o trabalho). Moreover, to increase the informativeness of this triple, it is possible to use some task of coreference. Thus the pronoun "nós" would be associated with the real subject (the people who are going to do the work).

6. Related Works

The precursor systems in Open IE performed their task in four stages: (i) Labeling of sentences using heuristics or distant supervised; (ii) An extractor training using a sequential labelling model (*e.g.* conditional random field (CRF)); (iii) Identification of the noun phrases (arguments); and (iv) the extractor identifies the relational phrase (if there is) between the arguments. The first proposals such as *TextRunner* [Banko et al. 2007] and *WOE* [Wu and Weld 2010] used this kind of approach. The use of machine learning was quickly replaced by methods that use heuristics or handcrafted rules for identifying relational phrases. New systems such as *ReVerb* [Fader et al. 2011] substituted the extractor by regular expression and started the task by first identifying the relational phrase to then discover its arguments. After the *ReVerb* that gave great pulse to the area, new methods emerged as *OLLIE* [Schmitz et al. 2012] and *ClausIE* [Del Corro and Gemulla 2013], but all exclusively for English texts. To the best of our knowledge, the first Open IE system for Portuguese texts was the *DepOE* [Gamallo et al. 2012] published in 2012. Although presenting a multilingual proposal, including Spanish and Galician, the evaluation of the method is only made for facts extracted in English sentences. Only in 2015 is published a new version of *DepOE* called *ArgOE* [Gamallo and Garcia 2015] where their evaluation for Portuguese was completed. This method uses handcrafted rules on a dependency tree to identify useful parts in sentences (clauses) and then uses the clauses to compose the extracted facts. Open IE with DP approach and handcrafted rules has been receiving greater attention because of their higher *recall*. On the other hand, for texts in Portuguese, there are methods that use shallow analysis. In 2013, *LSOE* [Xavier et al. 2013] published a work that uses POS tagger and a set of regular expressions identifying facts in an unsupervised way. This approach is similar to the method proposed by Sena *et al.* (2017). In this case, the syntactic restrictions of the *ReVerb* were used in Portuguese sentences. Furthermore, new syntactic restrictions were proposed to identify transitivity and symmetry to generate new facts and increase the *recall*. In Collovini *et al.* (2016) a method that uses CRF is presented to Portuguese. However, their extraction is limited to a restricted set of named entities. We believe that this limitation restricts the use of this method in specific applications, as listed by the authors.

The application of Open IE for Portuguese texts is scarce. The *RAPPort* system [Rodrigues and Gomes 2015] have proposed a method that uses DP and a standard subject-predicate-object to extract facts. The facts extracted by the Open IE module are used to enhance a QA system. In the best of our knowledge, other applications such as ontology construction, classification or clustering texts have not been identified. Thus, our proposal is positioned in this scenario of evolution from the Open IE to the Portuguese language.

7. Conclusions and Future Work

In this work, we proposed the *DependentIE*, an open IE for Portuguese texts. Our proposal differs from the state of the art by not using fixed manual rules. This feature enhanced our method when compared with another system of the same approach. To go through the dependency tree, we adapt an in-depth search that identifies the arguments of a relationship. The *DependentIE* was compared to the Open IE *ArgOE* and *InferReVerbPT* systems, obtaining greater gain under two different datasets: journalistic and encyclopedia texts. We believe that linguistic tools to a target language can increase this type of task. Our results suggested that this hypothesis is correct, but also reveals that Open IE for Portuguese still requires some efforts. These efforts go beyond the adaptation of existing approaches for other languages and the construction and improvement of tools for natural language processing.

As future works we need to enhance the DP tool for a more accurate *precision* and *recall*. We observed that our method did not extract facts from a significant number of sentences due to not identifying the subject on these sentences. There are also particular issues of the Portuguese language as an implicit or indeterminate subject and the anaphoras. All these concerns are current open problems. We believe that minimizing these problems can further enhance our results.

References

- Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *Proceedings of IJCAI*, volume 7, pages 2670–2676.
- Bast, H. and Hausmann, E. (2013). Open information extraction via contextual sentence decomposition. In *Proceedings of ICSC*, pages 154–159. IEEE.
- Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic. *Computational linguistics*, 22(2):249–254.
- Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.
- de Abreu, S. C., Bonamigo, T. L., and Vieira, R. (2013). A review on relation extraction with an eye on portuguese. *Journal of the Brazilian Computer Society*, 19(4):553–571.
- Del Corro, L. and Gemulla, R. (2013). Clausie: clause-based open information extraction. In *Proceedings of WWW*, pages 355–366. ACM.
- Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of EMNLP*, pages 1535–1545. Association for Computational Linguistics.
- Gamallo, P. and Garcia, M. (2015). Multilingual open information extraction. In *Proceedings of EPIA*, pages 711–722. Springer.
- Gamallo, P., Garcia, M., and Fernández-Lanza, S. (2012). Dependency-based open information extraction. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP, ROBUS-UNSUP '12*, pages 10–18. Association for Computational Linguistics.

- Nivre, J., Hall, J., and Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6, pages 2216–2219.
- Rodrigues, R. and Gomes, P. (2015). Rapport—a portuguese question-answering system. In *Portuguese Conference on Artificial Intelligence*, pages 771–782. Springer.
- Rodríguez, J. M., Merlino, H. D., Pesado, P., and García-Martínez, R. (2016). Performance evaluation of knowledge extraction methods. In *Proceedings of IEA/AIE*, pages 16–22. Springer.
- Schmitz, M., Bart, R., Soderland, S., Etzioni, O., et al. (2012). Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics.
- Sena, C. F. L., Glauber, R., and Claro, D. B. (2017). Inference approach to enhance a portuguese open information extraction. In *Proceedings of the 19th International Conference on Enterprise Information Systems - Volume 1: ICEIS*, pages 442–451. INSTICC, ScitePress.
- Wu, F. and Weld, D. S. (2010). Open information extraction using wikipedia. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 118–127. ACL.
- Xavier, C. C., de Lima, V. L. S., and Souza, M. (2013). Open information extraction based on lexical-syntactic patterns. In *Intelligent Systems (BRACIS), 2013 Brazilian Conference on*, pages 189–194. IEEE.
- Xavier, C. C., de Lima, V. L. S., and Souza, M. (2015). Open information extraction based on lexical semantics. *Journal of the Brazilian Computer Society*, 21(1):1.

Rumo à integração de múltiplos QAs de domínio restrito: sistema de classificação de domínio através das perguntas

Leandro Luciani Tavares¹, Tiago Agostinho Almeida¹, Renato Moraes Silva¹

¹Departamento de Computação (DComp)
Universidade Federal de São Carlos (UFSCar) – Sorocaba – SP – Brasil

leandro.ltavares@gmail.com, talmeida@ufscar.br,
renatoms@dt.fee.unicamp.br

Abstract. *Question Answering (QA) systems can be grouped into two categories: open domain and restricted one. Ideally, a QA system should combine the amplitude of an open domain QA with the depth of a restricted one. A possibility would be integrate multiple restricted domain QA to work together, in a seamless way to the user, as one open domain QA. Therefore, this paper presents a system for automatic selecting the domain of a question, which can be used to elect the QA system to answer a given question. Applying a synthetic question generator over a corpus, a set of millions of questions have been created and used to train a classifier able to predict the question domain. The performance of the proposed method was evaluated using human-elaborated questions and the results suggest that it is a very promising proposal and it can be used in the QA system selection task.*

Resumo. *Sistemas de Question Answering (QA) podem ser divididos em duas categorias: de domínio geral e de domínio restrito. Idealmente, um QA deveria ter a amplitude de um sistema de domínio geral e a profundidade de um restrito. Uma possibilidade seria integrar múltiplos QAs de domínio restrito para atuarem juntos, de forma transparente ao usuário, como um QA de domínio geral. Portanto, este artigo propõe um sistema de categorização automática de perguntas em domínios específicos, que pode ser utilizado para eleger o QA que será responsável pela elaboração da resposta. A partir do corpus de documentos, foi empregado um gerador de perguntas sintéticas para produzir milhares de amostras sobre as quais treinou-se um classificador para prever o domínio de perguntas feitas por usuários. O modelo proposto foi avaliado usando um conjunto de perguntas reais e os resultados indicam que a proposta é bastante promissora e que pode ser usada na prática para selecionar sistemas de QA.*

1. Introdução

Sistemas de QA (*Question Answering*) são aplicações cujo objetivo é oferecer respostas em linguagem natural para perguntas realizadas também em linguagem natural.

Com a evolução no modo de comunicação pessoal e na capacidade dos dispositivos, sistemas de QA se proliferaram em diversas áreas, como atendimento ao consumidor, assistentes pessoais e suporte ao aprendizado. Eles são empregados para atender rapidamente solicitações dos usuários, objetivando serem mais efetivos que operadores humanos.

Nesse contexto, recentemente surgiram sistemas extremamente avançados como o IBM Watson¹ e Inquire Biology [Chaudhri et al. 2015], frutos da enorme quantidade

¹What is Watson?, disponível em: <http://goo.gl/RTg5Hb>. Acessado em: 03/06/2017.

de informação digital disponível, alto poder computacional e evolução das técnicas de processamento de linguagem natural.

De maneira geral, os sistemas de QA realizam as seguintes etapas [Athira et al. 2013]: (i) processamento e análise da pergunta; (ii) recuperação de informação relevante contida em conjuntos de documentos, páginas, artigos, notícias e outras fontes de texto; e (iii) geração ou extração da resposta.

De acordo com [Allam e Haggag 2012], os sistemas de QA podem pertencer a uma das categorias a seguir:

1. **Domínio geral:** visa responder perguntas sobre diversas áreas de conhecimento sem se aprofundar em nenhuma delas e, conseqüentemente, costumam oferecer respostas sucintas e objetivas;
2. **Domínio restrito:** capaz de produzir respostas mais elaboradas, porém específicas a apenas uma ou poucas áreas de conhecimento. São conhecidos também como *RDQA (Restricted Domain Question Answering)*.

Idealmente, um sistema de QA deveria ter a *amplitude* de um sistema de domínio geral e a *profundidade* de um de domínio restrito. Nesse sentido, conforme sugerido por [Chung et al. 2004], diversos RDQAs poderiam ser combinados para formar um sistema de QA de domínio geral de forma transparente ao usuário. A Figura 1 apresenta uma visão geral dessa arquitetura.

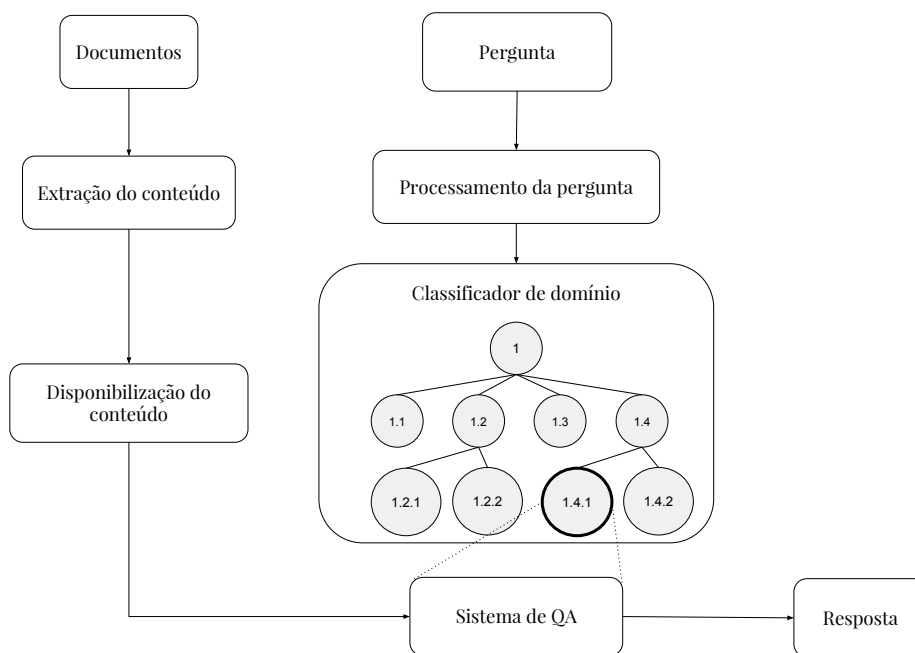


Figura 1. Arquitetura macro de um sistema de QA de domínio geral formado pela combinação de sistemas de QA de domínio restrito. Dada uma pergunta, um classificador de domínio é responsável por escolher adequadamente o sistema de QA de domínio restrito que irá elaborar a resposta.

Uma peça fundamental nesse modelo é o sistema de classificação de domínio, responsável por escolher corretamente o QA que irá elaborar a resposta para uma dada pergunta. Contudo, para treinar esse classificador, é mandatório a existência e atualização frequente de uma base de dados rotulada composta por amostras de qualidade e com ampla cobertura de diferentes domínios. Neste contexto, este trabalho apresenta uma

maneira simples, automática e eficiente de obter esse classificador, evitando o elevado custo da criação e atualização dessa base. Com isso, é possível oferecer aprendizado incremental e contínuo ao classificador e, assim, assegurar a eficiência geral do sistema de QA ao longo do tempo.

O restante desse trabalho está organizado da seguinte forma: a Seção 2 apresenta brevemente os trabalhos correlatos existentes na literatura. O sistema proposto para classificação de domínio é detalhado na Seção 3. Na Seção 4, é descrita a metodologia experimental empregada na avaliação da proposta. A Seção 5 apresenta os resultados obtidos. Por fim, a Seção 6 oferece conclusões e direcionamentos para trabalhos futuros.

2. Trabalhos relacionados

As pesquisas na área de QA foram iniciadas por [Simmons 1965], que discutiu a viabilidade de um computador responder perguntas em linguagem natural. Os primeiros sistemas de QA foram apresentados por [Green et al. 1961] e [Woods 1973], ambos sendo NLDBs (*Natural Language Interfaces for Databases*) e considerados os primeiros RDQAs por abordarem apenas domínios específicos.

As primeiras iniciativas de utilizar documentos textuais como fontes de dados para os sistemas de QA surgiram com o QUALM [Lehnert 1977], enquanto que o MURAX [Kupiec 1993] é considerado um dos pioneiros ao se tratar de sistemas de QA de domínio geral. Esse último, utilizava as Enciclopédias Grolier² como fonte de informação e, portanto, abordava diversos domínios.

Desde então, com o incentivo de conferências como o *TREC*³ (*Text REtrieval Conference*), surgiram diversos estudos que contribuíram com a evolução dos sistemas de QA, como por exemplo: [Clarke e Terra 2003], [Chung et al. 2004], [Minock 2005], [Mollá e Vicedo 2007], [Salloum 2009], [Miao et al. 2010] e mais recentemente por [Chaudhri et al. 2015]. Desses, destaca-se o trabalho apresentado por [Chung et al. 2004] que propôs o mecanismo de categorização da pergunta através de um classificador de domínio. A categorização de perguntas foi utilizada para determinar instâncias mais específicas de sistemas de QA, almejando com isso aumentar a qualidade das respostas e o desempenho geral do sistema.

Segundo [Song et al. 2011], categorização de perguntas é uma especialização da tarefa de classificação de textos curtos. De maneira geral, na literatura são encontradas duas abordagens principais: detecção da categoria (domínio) [Song et al. 2011] e identificação do tipo da pergunta [Ittycheriah et al. 2001, Li e Roth 2002, Zhang e Lee 2003].

Quanto à categorização do domínio de documentos e perguntas, inicialmente foram apresentadas técnicas baseadas em regras criadas a partir de notícias [Hayes e Weinstein 1990]. Mais recentemente, outras propostas foram apresentadas, sendo que grande parte delas são baseadas em aprendizado de máquina, utilizando métodos tradicionais de classificação, como: *naïve* Bayes [Koller e Sahami 1997, McCallum et al. 1998], máquinas de vetores de suporte [Joachims 1998], *K*-vizinhos mais próximos [Yang e Liu 1999] e regressão logística [Zhang e Oles 2001]. Em [Zhang e Lee 2003], por exemplo, o método de máquinas de vetores de suporte foi empregado para categorizar perguntas utilizando um conjunto pré-definido de amostras rotuladas e disponibilizadas pelo *TREC*³. [Song et al. 2011] apresentaram uma abordagem automática de categorização de perguntas que envolve a construção do espaço vetorial

²Enciclopédias Grolier, disponível em <http://goo.gl/J7sdsU>. Acessado em 29/06/2017.

³*Text REtrieval Conference (TREC)*, disponível em: <http://trec.nist.gov>. Acessado em 21/06/2017.

de atributos, identificação de termos relevantes, mapeamento semântico para inserção e substituição de atributos e cálculo de similaridade entre a pergunta e cada categoria.

3. Sistema de classificação de domínio

Um dos principais e também mais custosos componentes da arquitetura de QA apresentada na Figura 1 é o classificador de domínios. O seu processo de criação e manutenção envolve etapas trabalhosas e delicadas como a coleta, pré-processamento, seleção, rotulação manual e atualização de um conjunto representativo de amostras.

A qualidade geral do sistema de QA depende diretamente do desempenho do classificador de domínio que é responsável por escolher a instância de RDQA que irá responder uma dada pergunta. Assim, é desejável que a base de dados de treinamento seja ampla em termos de contextos sobre domínios distintos e pré-definidos. Por exemplo, no domínio de “esportes”, é recomendável que existam amostras de treinamento que abordem diferentes contextos, tais como a história dos mais variados esportes, recordes, atletas, competições e resultados, de tal forma que o sistema seja robusto em relação a grande diversidade de perguntas possíveis que um usuário possa fazer. Além disso, para evitar a defasagem com o tempo, é mandatário que a atualização do classificador ocorra de maneira incremental, sendo ele treinado constantemente com amostras que abordam assuntos recentes.

Sistemas de QA em geral baseiam-se em grandes *corpus* de documentos e ontologias para identificar fragmentos dos quais seja possível extrair respostas para as perguntas [Mollá e Vicedo 2007]. Contudo, não é recomendável utilizar diretamente os próprios documentos do *corpus* para treinar um classificador de domínio de perguntas, pois a representação das amostras é muito diferente. Para ilustrar, considere a seguinte pergunta: “*Who is a pioneer in cognitive sciences?*”. O fragmento de texto⁴ apresentado a seguir contém a resposta destacada em negrito: “*Avram Noam Chomsky (born December 7, 1928) is an American linguist, philosopher, cognitive scientist, historian, social critic, and political activist. Sometimes described as “the father of modern linguistics”, Chomsky is also a major figure in analytic philosophy and one of the founders of the field of cognitive science...*”. É importante destacar as importantes diferenças entre a pergunta e o fragmento de texto:

1. **Dimensão:** a pergunta é composta por apenas 7 palavras, enquanto o fragmento da resposta possui algumas dezenas e o texto original completo é composto por mais de 10 mil. Portanto, a pergunta geralmente é representada por um vetor muito mais esparsa que o texto original e ambos os vocabulários são diferentes em qualidade e principalmente em quantidade de atributos;
2. **Entidades:** conforme apresentado por [Ittycheriah et al. 2001], esta informação pode ser obtida exclusivamente das perguntas e indica o tipo de entidade esperada como resposta. No exemplo, o pronome interrogativo “*Who*” indica que a entidade esperada na resposta é uma pessoa;
3. **Sinônimos:** na pergunta, foi usado o termo “*pioneer*”, enquanto que no texto aparece o sinônimo “*founder*”.

Com base nessas diferenças, é esperado que um classificador treinado com amostras que representam perguntas venha a obter desempenho muito superior na classificação de domínio de perguntas do que um modelo treinado com os documentos originais. Contudo o processo de criação e manutenção de uma base de dados rotulada consumiria muito tempo e poderia ser impeditivo para a geração desse classificador. Nesse contexto, neste trabalho é proposto um modelo para automatizar o processo de geração de perguntas e

⁴Noam Chomsky, disponível em: <http://goo.gl/teoGsA>. Acessado em: 03/06/2017

oferecer treinamento contínuo a um classificador de domínio. Esse modelo, ilustrado na Figura 2, consiste na criação de amostras sintéticas a partir dos documentos que compõem o próprio *corpus* utilizado para aquisição de conhecimento dos RDQAs.

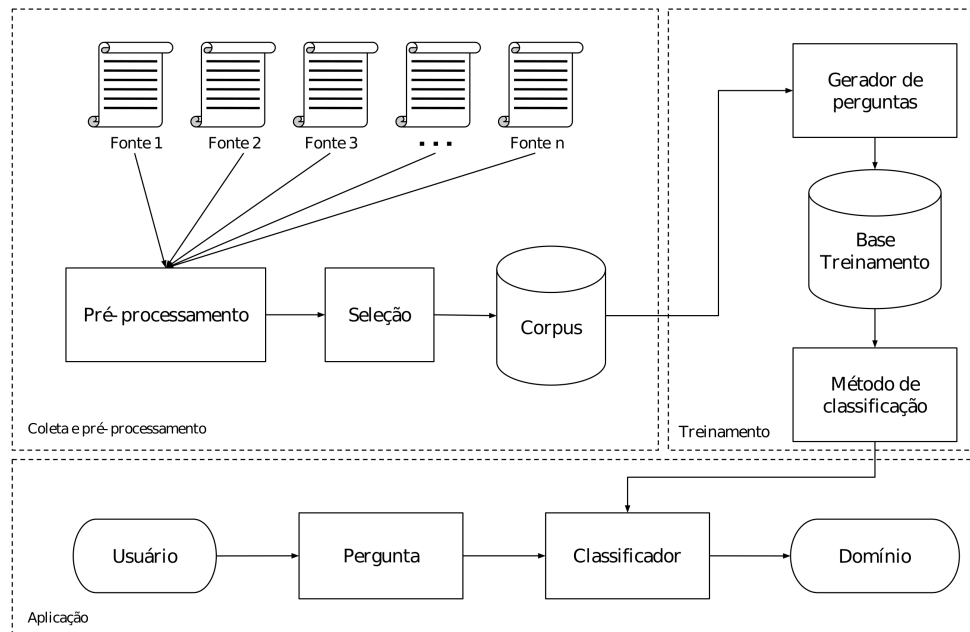


Figura 2. Diagrama do modelo proposto: os documentos que compõem o *corpus* são usados como entrada para uma etapa de geração de perguntas sintéticas que, por sua vez, são usadas para treinar o classificador de domínio.

No modelo proposto, os documentos que compõem o *corpus* são utilizados como entrada para um processo de *geração de perguntas sintéticas*, sendo que as classes usadas como rótulos correspondem às categorias (domínios) dos documentos originais utilizados na entrada.

Para realizar a geração automática das perguntas, foi utilizado o *Question Generation via Overgenerating Transformations and Ranking*⁵ [Heilman e Smith 2009]. Como há indícios de que as respostas para as perguntas potencialmente mais relevantes se encontram nas linhas iniciais do documento⁶, foram utilizados apenas os dois primeiros parágrafos de cada documento como entrada para o processo de geração.

No processo de geração, os fragmentos iniciais dos documentos são segmentados em frases, sobre as quais é aplicado um processo de transformação sintática. Através de um conjunto de regras bem definidas, incluindo o “*Wh-movement*” e o “*subject-auxiliary inversion*”, as afirmações são transformadas em perguntas [Heilman e Smith 2009].

O gerador automático foi configurado para descartar perguntas nas quais os pronomes não possam ser resolvidos, de forma a evitar perguntas do tipo: “*Who is she?*”. Além disso, também foi configurado para se descartar perguntas com mais de 30 termos e priorizar aquelas que iniciam com *Wh* (*Who, When, What, Where, ...*). A Tabela 1 apresenta alguns exemplos de perguntas sintéticas geradas a partir de um fragmento de uma notícia⁷.

⁵*Question Generation via Overgenerating Transformations and Ranking*, disponível em: <http://www.cs.cmu.edu/~ark/mheilman/questions/>. Acessado em: 03/06/2017.

⁶*Wikipedia Question Generator*, disponível em: <https://github.com/atbaker/>

Tabela 1. Exemplos de perguntas sintéticas geradas a partir de um fragmento de texto.

Tipo	Conteúdo
Fragmento original	<i>The renowned performance artist Marina Abramović danced with Jay-Z at a New York art gallery this week as part of the rapper's latest video shoot. It is the latest in a series of crossovers between art and pop that at first glance seem unlikely. As if Ai Weiwei and Anish Kapoor performing Gangnam Style and Weiwei also recording a heavy metal album were not enough, here is Abramović, a legend of contemporary art, dancing to Jay-Z's Picasso Baby. Meanwhile Yoko Ono's Plastic Ono Band got rave reviews in London this summer. (...) Abramović is in her late 60s, Ai Weiwei well into his 50s. Yoko Ono, one half of art and pop's ultimate marriage, is 80.</i>
Pergunta 1	<i>Who is one half of art and pop's ultimate marriage?</i>
Pergunta 2	<i>Who is a legend of contemporary art?</i>
Pergunta 3	<i>What does Yoko Ono's Plastic Ono Band's not just another example of?</i>
Pergunta 4	<i>Who danced with Jay-Z at a New York art gallery this week as part of the rapper's latest video shoot?</i>
Pergunta 5	<i>Who is 80?</i>

Como algumas perguntas geradas podem não possuir atributos suficientes para a identificação do domínio, como por exemplo: “*Who is 80?*”, foi empregado um processo automático de validação de cada pergunta. Para isso, são removidas temporariamente as *stopwords*⁸ e todos os caracteres não alfabéticos e, caso restem menos do que 3 termos, a pergunta é descartada. Além disso, perguntas duplicadas também não são armazenadas.

O gerador de perguntas é capaz de criar automaticamente dezenas ou mesmo centenas de amostras a partir de um documento de texto. Dessa forma, é possível obter facilmente uma base de treinamento de larga escala, composta por milhares de amostras rotuladas. Além disso, o processo de geração de perguntas pode acompanhar a atualização do *corpus*, sendo que novas amostras podem ser geradas quando documentos novos são inseridos no repositório de dados. Portanto, o classificador de domínio pode ser automaticamente atualizado de maneira incremental, de forma a manter e até melhorar o seu poder preditivo ao longo do tempo.

4. Metodologia experimental

Primeiramente, foi criado um *corpus* de documentos de texto composto por (i) artigos da Wikipedia anglófona⁹ e (ii) notícias dos seguintes portais e jornais *online*: BBC¹⁰, Fox News¹¹, NBC News¹², Wired¹³, The Guardian¹⁴ e The Wall Street Journal¹⁵, publicadas em língua inglesa entre 02/07/2004 e 02/10/2016. Portanto, o *corpus* é heterogêneo em sua composição e contém documentos de tamanhos variados e diferentes estilos de escrita.

Através de um *crawler*, foi extraído apenas o conteúdo textual das notícias, o título e a categoria, descartando-se a formatação do texto, assim como imagens, *links*, anúncios, comentários, referências e outros elementos não textuais. A mesma limpeza foi aplicada

wikipedia-question-generator. Acessado em: 03/06/2017.

⁷Jay-Z and Marina Abramović rekindle art's relationship with pop, disponível em: <http://goo.gl/DdCicq>. Acessado em 29/06/2017.

⁸As *stopwords* removidas foram aquelas presentes no repositório NLTK, disponível em: <http://www.nltk.org/book/ch02.html>. Acessado em: 30/06/2017.

⁹Wikipedia Dumps, disponível em: <http://goo.gl/peMnDM>. Coletado em 20/11/2016.

¹⁰BBC, disponível em: <http://www.bbc.com>. Acessado em 21/06/2017.

¹¹Fox News, disponível em: <http://www.foxnews.com>. Acessado em 21/06/2017.

¹²NBC News, disponível em: <http://www.nbcnews.com>. Acessado em 21/06/2017.

¹³Wired, disponível em: <http://www.wired.com>. Acessado em 21/06/2017.

¹⁴The Guardian, disponível em: <http://goo.gl/w7jcrP>. Acessado em 21/06/2017.

¹⁵The Wall Street Journal, disponível em: <http://www.wsj.com>. Acessado em 21/06/2017.

sobre os artigos da *Wikipedia*. O repositório coletado totalizou 6.660.814 documentos, dos quais 1.432.453 são notícias e 5.228.361 são artigos da *Wikipedia*.

Em seguida, as categorias dos documentos foram normalizadas utilizando a taxonomia definida pelo IPTC (*International Press Telecommunications Council*)¹⁶, sendo selecionados apenas os documentos pertencentes aos seguintes domínios: [Arts & Entertainment & Culture], [Economy & Business & Financial], [Politics], [Science & Technology] e [Sport]. Após essa normalização, o *corpus* de documentos pertencentes às categorias de interesse totalizou 460.048, sendo 123.932 notícias e 336.116 artigos da *Wikipedia*. A Tabela 2 apresenta a distribuição dos documentos em cada categoria.

Tabela 2. Quantidade de documentos por categoria.

Categoria	Notícias	Wikis
Arts & Entertainment & Culture	48.860	152.963
Economy & Business & Financial	17.387	14.039
Politics	23.690	44.898
Science & Technology	10.450	24.284
Sport	23.545	99.932
Total	123.932	336.116

Cada um desses documentos foi utilizado como entrada para o gerador de perguntas e, no final desse processo, foram criadas automaticamente **3.517.858** perguntas sintéticas rotuladas. Esse conjunto será chamado de *Tr* no decorrer deste trabalho.

Na fase de treinamento do classificador de domínio, as amostras de *Tr* foram primeiramente convertidas para letras minúsculas e segmentadas em múltiplos termos utilizando como delimitadores quaisquer caracteres não alfanuméricos. A representação utilizada foi o tradicional modelo espaço-vetorial (*bag of words*) e os valores usados para representar os termos foram a frequência de aparecimento em cada pergunta.

Devido ao volume elevado de amostras e à necessidade de aprendizado incremental e de velocidade na classificação, foi empregado o método de classificação *naïve Bayes* multinomial, que é amplamente utilizado para categorização de textos [McCallum et al. 1998, Nguyen et al. 2011]. Ele foi implementado usando a linguagem de programação Python, através da função disponibilizada pela biblioteca `scikit-learn`¹⁷.

Levando-se em conta que as perguntas sintéticas podem possuir características distintas das perguntas reais feitas por humanos, foi criada uma base de teste composta somente por perguntas reais, extraídas do portal *Quora*¹⁸, distribuídas nas mesmas categorias de documentos do *corpus* (categorias listadas na Tabela 2). Como as perguntas do *Quora* são categorizadas automaticamente por um método de classificação proprietário, elas passaram por um processo de validação manual para assegurar que foram corretamente rotuladas. Após esse processo, restaram 500 perguntas na base de teste (*Te*).

Para comparar os resultados, foram utilizadas as seguintes medidas de desempenho tradicionalmente utilizadas em tarefas de classificação:

- **acurácia**: razão entre o total de acertos e o total de predições; e
- **F-medida**: média harmônica entre a precisão e a revocação (*recall*).

A Figura 3 sumariza a metodologia experimental empregada neste trabalho e a Tabela 3 apresenta as estatísticas básicas sobre as bases de dados *Tr* e *Te*, tais como a

¹⁶*Select and Show IPTC NewsCode Taxonomies as Tree Diagram*, disponível em: <http://show.newscodes.org/index.html?newscodes=subj>. Acessado em 03/06/2017.

¹⁷*Scikit learn*, disponível em <http://scikit-learn.org/stable/>. Acessado em 23/06/2017.

¹⁸*Quora*, disponível em: <http://www.quora.com>. Acessado em 21/06/2017.

quantidade de amostras por categoria, o tamanho do vocabulário, além da média, desvio padrão, mediana e a amplitude interquartil do número de termos por amostra.

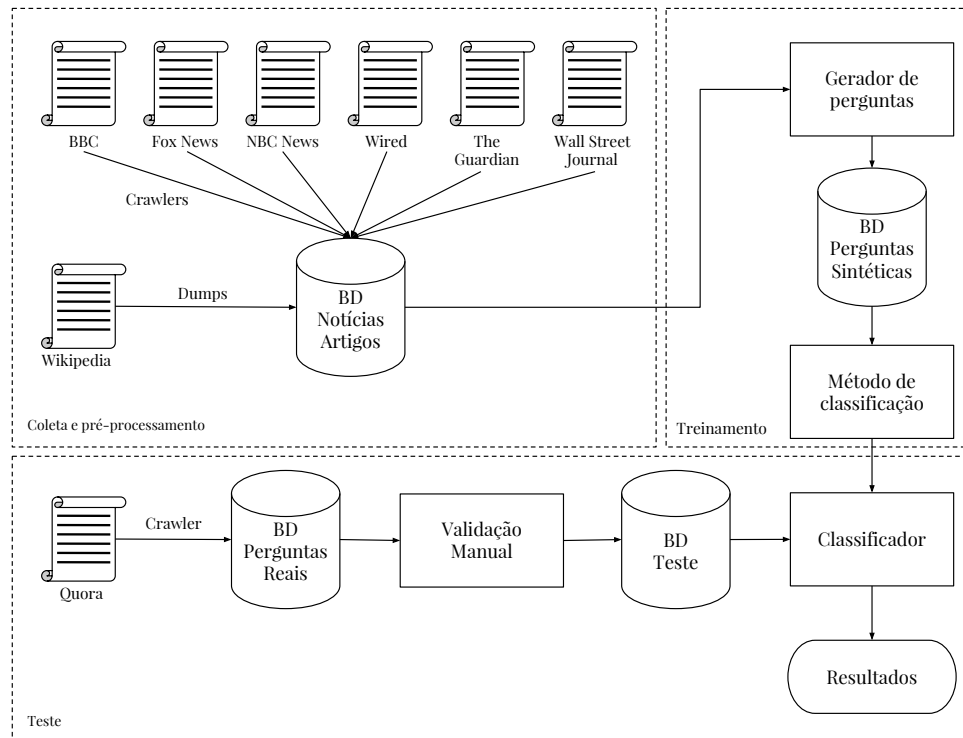


Figura 3. Diagrama da metodologia experimental empregada. Primeiramente, foram coletados, pré-processados e selecionados os documentos que compõem o *corpus*. Em seguida, cada um desses documentos foi usado como entrada para a etapa de geração de perguntas sintéticas que, por sua vez, foram usadas para treinar o classificador de domínio. Para avaliar o desempenho desse classificador, foi utilizado um conjunto de teste composto por perguntas reais.

Tabela 3. Estatísticas da base de dados de treinamento (T_r) e de teste (T_e).

Informações gerais			Quantidade de amostras por categoria		
	T_r	T_e	Categoria	T_r	T_e
Tamanho do vocabulário	228.643	2.004	Arts & Entertainment & Culture	1.305.860	100
Média	10.9	13.3	Economy & Business & Financial	265.196	100
Desvio padrão	5.3	5.2	Politics	262.284	100
Mediana	10.0	12.0	Science & Technology	466.199	100
Amplitude interquartil	7.0	7.0	Sport	1.218.319	100
			Total	3.517.858	500

É evidente que o tamanho do vocabulário de T_r é muito maior que o de T_e , o que é ocasionado pela grande diferença na quantidade de amostras entre essas bases. Contudo, a quantidade média de termos que compõe cada amostra é baixa e similar em ambas. Isso indica que cada amostra é representada por um vetor altamente esparsa.

Com o intuito de permitir a reprodução dos experimentos, as bases de dados T_r e T_e estão disponíveis publicamente para consulta¹⁹.

¹⁹Labeled Documents and Questions Dataset, disponível em: <http://lasid.sor.ufscar.br/questionclassifier/qc/moreinfo#datasets>. Acessado em 25/06/2017.

5. Resultados

Foram realizados experimentos considerando três cenários de treinamento:

- **Cenário 1:** utilizando a base de dados original (Tr);
- **Cenário 2:** utilizando Tr após remoção de *stopwords*; e
- **Cenário 3:** utilizando Tr após remoção de *stopwords* e balanceamento das categorias através de subamostragem aleatória [He e Garcia 2009].

O classificador gerado em cada cenário foi empregado para predizer os rótulos das amostras de teste (Te). Os resultados obtidos estão sumarizados na Tabela 4.

Tabela 4. Resultados obtidos na classificação de domínio.

	Cenário 1	Cenário 2	Cenário 3
Categoria	F-medida		
Arts & Entertainment & Culture	0,775	0,795	0,847
Economy & Business & Financial	0,816	0,838	0,847
Politics	0,582	0,688	0,800
Science & Technology	0,767	0,802	0,819
Sport	0,925	0,931	0,954
Média	0,773	0,811	0,853
Acurácia	0,784	0,816	0,854

De maneira geral, o classificador de domínio obteve bom desempenho em todos os cenários avaliados, com F-medida média superior a 0,77. Contudo, é evidente que o classificador obteve melhor desempenho em certas categorias que em outras. Por exemplo, em “Sport”, a F-medida superou 0,95 no cenário 3, enquanto que “Politics” foi mais desafiadora, principalmente considerando os cenários 1 e 2. Um dos fatores que influenciou o desempenho dos métodos nas diferentes classes foi o desbalanceamento. Porém, mesmo após balancear a quantidade de amostras entre as classes (cenário 3), ainda foi constatada uma diferença de desempenho. Isso provavelmente deve-se à variedade de contexto possível inerente às amostras de cada classe. Por exemplo, vários termos relacionados às perguntas da classe “Politics” também podem ser usados nas classes “Economy & Business & Financial” e “Science & Technology”. Por outro lado, as amostras da classe “Sport” costumam usar termos mais específicos, pouco presentes em outras classes.

Comparando os cenários 1 e 2, é notório que a remoção de *stopwords* foi benéfica, pois o desempenho do classificador melhorou em todas as categorias. Além disso, o balanceamento entre as classes também favoreceu o poder preditivo e permitiu que o classificador atingisse acurácia superior a 85% e F-medida média maior que 0,85. Tal desempenho indica que o modelo proposto é promissor e que pode ser aplicado com sucesso na prática, dada a facilidade de criação automática de amostras e atualização do modelo de predição.

Como o custo computacional do treinamento é diretamente proporcional à dimensionalidade da base de dados e, neste caso, o número de termos no vocabulário da base Tr é elevado (228.643), foi também avaliado o impacto da seleção de atributos na capacidade preditiva do classificador. Para isso, considerando o cenário 2, foi usado o método de fatores de confiança (CF – *confidence factors*) [Assis et al. 2006] para selecionar os 5.000, 10.000, 50.000, 100.000 e 150.000 termos mais representativos. Conforme apresentado na Figura 4, a seleção de atributos é prejudicial ao desempenho (indicado pela F-medida média) na tarefa de classificação.

O classificador de domínio proposto e apresentado neste trabalho está disponível publicamente para ser utilizado de maneira *online*²⁰. A aplicação disponibilizada prediz

²⁰*Question Domain Classifier*, disponível em: <http://lasid.sor.ufscar.br/questionclassifier/qc/>. Acessado em 25/06/2017.

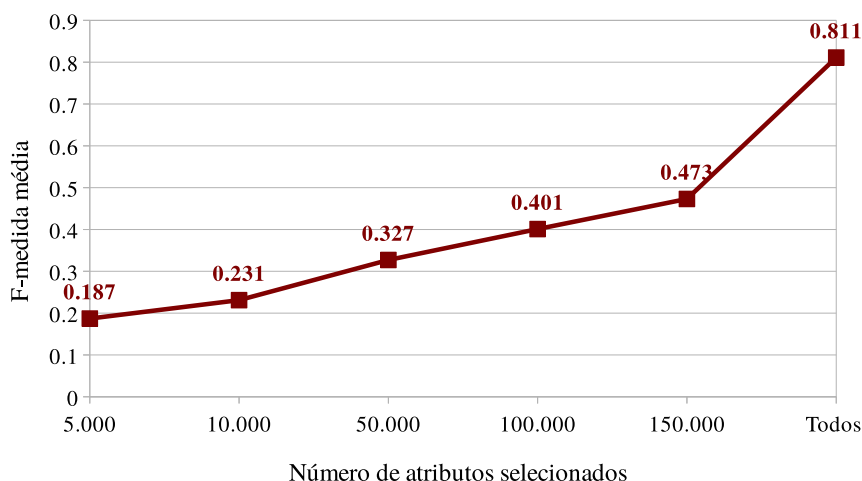


Figura 4. F-medida média obtida na classificação de domínio das perguntas variando a quantidade de atributos.

o domínio de qualquer pergunta escrita em inglês.

6. Conclusão

Sistemas de QA são amplamente utilizados e vêm ganhando cada vez mais notoriedade nos dias atuais, principalmente devido ao acelerado crescimento no volume de dados digitais. Idealmente, um QA deveria ter a amplitude de um sistema de domínio geral e a profundidade de um de domínio restrito. Para isso, é possível integrar múltiplos QAs de domínio restrito para atuarem juntos, de forma transparente ao usuário, como um QA de domínio geral. Nesse caso, é necessário empregar um sistema de classificação de domínio para selecionar o QA que deverá produzir a resposta. Portanto, a qualidade geral do sistema de QA depende diretamente do desempenho do classificador de domínio.

A etapa mais custosa e delicada dentre os processos de criação e manutenção de um sistema de classificação de domínio reside na coleta, pré-processamento, seleção, rotulação manual e atualização de um conjunto representativo de amostras. Assim, é desejável que a base de dados de treinamento seja ampla, atualizada e representativa em termos de amostras com diferentes contextos.

Este artigo propôs uma forma de treinar esse classificador a partir de uma base de dados criada de maneira automática a partir do próprio *corpus* de documentos usado pelo sistema de QA. Para isso, foi utilizado um gerador de perguntas sintéticas que foi capaz de produzir automaticamente milhares de amostras sobre as quais treinou-se um classificador para prever o domínio de perguntas. O desempenho do modelo proposto foi avaliado usando um conjunto de perguntas reais e os resultados obtidos, com acurácia superior a 85%, indicam que a proposta é promissora e pode ser usada na prática.

De maneira geral, os resultados também evidenciaram que as amostras sintéticas possuem qualidade suficiente para obter bons resultados na predição em aplicações reais. A vantagem desse modelo reside na economia de tempo e esforço que seriam necessários para obter e manter atualizada uma significativa quantidade de perguntas reais e rotuladas para treinar o método de classificação de domínio.

Atualmente, está sendo desenvolvido o sistema de QA de domínio geral composto por múltiplas instâncias de domínio restrito conforme apresentado na Figura 1. Em trabalhos futuros, planeja-se estender a classificação de domínio para um modelo hierárquico de forma a representar as perguntas e os documentos seguindo a taxonomia natural do

conteúdo, agregando um nível de detalhamento maior na classificação das categorias das perguntas, de forma a permitir selecionar o sistema de QA com maior precisão. Pretende-se também abordar o problema como uma tarefa de classificação *multilabel* para mitigar casos de ambiguidade de classes entre os documentos.

Agradecimentos

Os autores são gratos à agência de fomento CAPES (Proc. 1709642) pelo apoio financeiro a este projeto de pesquisa.

Referências

- Allam, A. M. N. e Haggag, M. H. (2012). The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences*, 2(3):211–220.
- Assis, F., Yerazunis, W., Siefkes, C., e Chhabra, S. (2006). Exponential differential document count – a feature selection factor for improving Bayesian filters accuracy. In *Proceedings of the 2006 MIT Spam Conference (SP'06)*, pages 1–6, Cambridge, MA, USA.
- Athira, P., Sreeja, M., e Reghuraj, P. (2013). Architecture of an Ontology-Based Domain-Specific Natural Language Question Answering System. *International Journal of Web & Semantic Technology*, 4(4):31–39.
- Chaudhri, V. K., Overholtzer, A., e Spaulding, A. (2015). *An Intelligent Textbook that Answers Questions*, pages 131–135. Springer International Publishing, Cham.
- Chung, H., Song, Y.-I., Han, K.-S., Yoon, D.-S., Lee, J.-Y., Rim, H.-C., e Kim, S.-H. (2004). A Practical QA System in Restricted Domains. In *Proceedings of the 2004 Annual Meeting of the Association for Computational Linguistics (ACL' 04) - Workshop on Question Answering in Restricted Domains*, pages 39–45, Barcelona, Espanha.
- Clarke, C. L. A. e Terra, E. L. (2003). Passage retrieval vs. document retrieval for factoid question answering. In *Proceedings of the 2003 Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR'03)*, pages 427–428, New York, NY, USA. ACM.
- Green, B. F., Wolf, A., Chomsky, C., e Laughery, K. (1961). Baseball: An automatic question answerer. In *Proceeding of the 1961 Western Joint IRE-AIEE-ACM computer conference (IRE-AIEE-ACM '61)*, volume 19, pages 219–224. Western Computing Conference.
- Hayes, P. J. e Weinstein, S. P. (1990). CONSTRUE/TIS: A system for content-based indexing of a database of news stories. In *Proceedings of the The 2nd Conference on Innovative Applications of Artificial Intelligence (IAAI'90)*, volume 90, pages 49–64, Washington, D.C, USA.
- He, H. e Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.
- Heilman, M. e Smith, N. A. (2009). Question generation via overgenerating transformations and ranking. Technical report, DTIC Document.
- Ittycheriah, A., Franz, M., e Roukos, S. (2001). IBM's statistical question answering system-TREC-10. In *The 10th Text REtrieval Conference (TREC'2001)*, pages 258–264, Gaithersburg, Maryland.
- Joachims, T. (1998). Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML'98)*, pages 137–142, Chemnitz, Germany. Springer.

- Koller, D. e Sahami, M. (1997). Hierarchically classifying documents using very few words. In *Proceedings of the 1997 International Conference on Machine Learning (ICML '97)*, pages 170–178, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Kupiec, J. (1993). MURAX: A robust linguistic approach for question answering using an on-line encyclopedia. In *Proceedings of the 1993 annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'93)*, pages 181–190, Pittsburgh, PA, USA. ACM.
- Lehnert, W. (1977). Human and computational question answering. *Cognitive Science*, 1(1):47–73.
- Li, X. e Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics (COLING '02)*, volume 1, pages 1–7.
- McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *The Fifteenth workshop on learning for text categorization (AAAI'98)*, volume 752, pages 41–48. Madison, WI, USA.
- Miao, Y., Su, X., e Li, C. (2010). Improving question answering based on query expansion with Wikipedia. In *Proceedings of the 2010 IEEE International Conference on Tools with Artificial Intelligence (ICTAI '10)*, volume 2, pages 233–240, Arras, França. IEEE.
- Minock, M. (2005). Where are the ‘killer applications’ of restricted domain question answering. In *Proceedings of the IJCAI Workshop on Knowledge Reasoning in Question Answering (IJCAI'05)*, page 4, Edimburgo, Escócia.
- Mollá, D. e Vicedo, J. L. (2007). Question Answering in Restricted Domains: An Overview. *Computational Linguistics*, 33(1):41–61.
- Nguyen, H. M., Cooper, E. W., e Kamei, K. (2011). Online learning from imbalanced data streams. In *Proceedings of the 3rd International Conference of the Soft Computing and Pattern Recognition (SoCPaR' 11)*, pages 347–352, Dalian, China. IEEE.
- Salloum, W. (2009). A question answering system based on conceptual graph formalism. In *Proceedings of the 2009 International Symposium on Knowledge Acquisition and Modeling (KAM '09)*, volume 3, pages 383–386, Wuhan, China. IEEE.
- Simmons, R. F. (1965). Answering english questions by computer: A survey. *Communications of the ACM*, 8(1):53–70.
- Song, W., Wenyin, L., Gu, N., Quan, X., e Hao, T. (2011). Automatic categorization of questions for user-interactive question answering. *Information Processing & Management*, 47(2):147–156.
- Woods, W. A. (1973). Progress in natural language understanding: an application to lunar geology. In *Proceedings of National Computer Conference and Exposition (AFIPS'73)*, pages 441–450, New York, NY, USA. ACM.
- Yang, Y. e Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'99)*, pages 42–49. ACM.
- Zhang, D. e Lee, W. S. (2003). Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'03)*, pages 26–32. ACM.
- Zhang, T. e Oles, F. J. (2001). Text categorization based on regularized linear classification methods. *Information retrieval*, 4(1):5–31.

Bayesian Networks and Massey's Method applied to Soccer Outcome Prediction

Júlio César Teixeira de Oliveira¹, Edimilson Batista dos Santos¹

¹Department of Computer – Federal University of São João del-Rei (UFSJ)
São João del-Rei – MG – Brazil

juliocto2011@gmail.com, edimilson.santos@ufs.br

***Abstract.** The problem of modeling the outcomes of matches in association football (soccer) has been studied for several years. In this paper, it is presented a prediction model which applies the Bayesian network formalism and Massey ranking method. The model was used to generate forecasts about the outcomes of the Brazilian Championship A Series matches during season 2014. In the experiments, the proposed prediction model has presented promising results.*

1. Introduction

Soccer is the world's most popular sport and constitutes the fastest growing gambling market, according to [Constantinou et al. 2012]. Predicting the results of soccer matches is interesting to many, from fans to punters. It is also interesting as a research problem, in part due to its difficult, because the result of a soccer match is dependent on many factors, such as coaching strategy, technical capacity of players, cheerleading, teamwork, individual motivation, concentration, temperature, fatigue, psychological pressure, confidence, luck and so on. So even for experts, it is very hard to predict the exact results of individual matches. Then, the prediction of soccer matches is one important research field and has been studied for multiple decades now [Hoekstra 2012].

As alternative, researchers continue to introduce a variety of soccer models which are formulated by diverse forecast methodologies. The previous forecast models predict the amount of goals for each team and infer the actual winner of the match from this result. Other approaches determine the outcome (win, draw, loss) of the match directly. While some of these focus on predicting tournament outcomes or league positions, our interest is in predicting outcomes of individual matches.

In this scenario with so many uncertainties, the formalism of Bayesian networks [Pearl 1998] can be considered a good tool for forecasting outcomes of soccer matches. Bayesian networks belong to the family of probabilistic graphical models and generally are used to represent uncertain knowledge in a particular domain. Thus, in this paper, we present a prediction model which applies Bayesian networks for predicting soccer match outcomes (win-draw-loose). Since soccer matches have the highly stochastic nature, we also propose to use Massey ranking method [Langville 2012] in our model to improve

the probability of the prediction. Massey's method has been used by the Bowl Championship Series¹ (or BCS) system to select NCAA football bowl matchups.

The remainder of this paper is organized as follows: Section 2 presents a background on Bayesian networks and Massey's method. Section 3 gives a summary of previous work on football prediction. In Section 4, we describe our prediction model, explaining how our Bayesian network and Massey's method interact. In Section 5, we present our experimental evaluation and analysis. Finally, conclusions and future works are given in Section 6.

2. Background

2.1. Bayesian Networks

Bayesian networks are graphical representations of probability distributions and they have been very used to represent uncertainty in Artificial Intelligence. According to [Heckerman et al. 1995], Bayesian networks represent an important role in modern expert systems, diagnosis machines and decision support systems.

A Bayesian network G [Pearl 1998] consists of two important components:

- i) A directed acyclic graph (DAG) structure. Each node in the graph represents a discrete random variable in the domain. An edge, $Y \rightarrow X$ in the graph, describes a parent and child relationship where Y is the parent and X is the child. All parents of X constitute the parent set of X , which is denoted by π_X .
- ii) In addition to the graph, each node X has an associated conditional probability table (CPT) specifying the probability of each possible state of X given each possible combination of states of the nodes in π_X . If a node X is such that $\pi_X = \emptyset$, its associated table gives the marginal probabilities of X . The probability tables are updated by using Bayes' Theorem, based on new sample information [Pearl 1998].

Fig. 1 presents an example of Bayesian network and shows the direct relation among its variables, which are connected by oriented edges. We observe that x_i is directly related to x_j if there is an edge from x_i to x_j . The joint probability distribution is represented as the product of conditional probability distributions. Thus, we have the following formula (1) for the Bayesian network of the Fig. 1:

$$P(x_1, x_2, \dots, x_6) = P(x_6|x_5)P(x_5|x_2x_3)P(x_4|x_1x_2)P(x_3|x_1)P(x_2|x_1)P(x_1) \\ = \prod_{i=1}^m P(x_i | \pi_{x_i}) \quad (1)$$

where m is the number of variables; $P(x_1, x_2, \dots, x_6)$ is the joint probability of $X = \{x_1, x_2, \dots, x_6\}$, which is a set of random variables; $P(x_i|x_j)$ is the conditional probability of x_i , given x_j , and π_{x_i} is the set of parents of x_i .

¹ "The Bowl Championship Series (BCS) is a rating system for NCAA college football that was designed to determine which teams are invited to play in which bowl games. The BCS has become famous, and perhaps notorious, for the ratings it generates for each team in the NCAA." [Langville 2012]

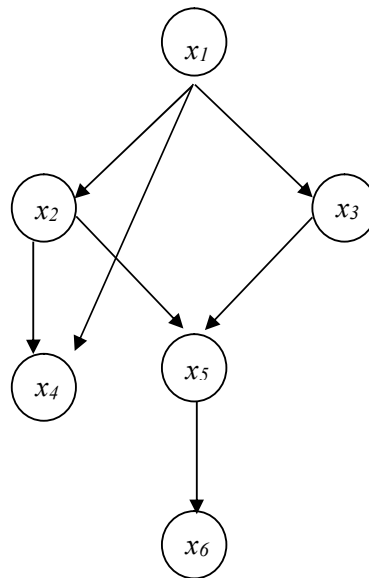


Figure 1. Example of Bayesian network.

Once the Bayesian network is defined, the knowledge represented on it can be drawn through a process called inference. The Bayesian inference calculates the conditional probability distribution from a query variable set, taking in account the values of an evidence variable set. The readers more interested in this issue can consult [Neapolitan 2003].

2.1. Massey Ranking Method

According to Langville (2012), the problem of ranking is elegantly simple yet some of its solutions can be complicated and full of paradoxes and conundrums. It can be defined as: arrange a group of items in order of importance.

One reason for the increased interest on the ranking problem comes from today's data collection capabilities. There is no shortage of interesting and real datasets in nearly every field imaginable from the real estate and stock markets to sports and politics to education and psychology. Our interest here is at ranking sports teams.

There are many ranking techniques which can be studied in [Langville 2012]. The ranking method used in our prediction model is the Massey method [Langville 2012].

Massey method was created by Kenneth Massey for ranking college football teams and it uses the mathematical theory of least squares. Thus, the fundamental philosophy of the method can be summarized with one idealized equation (2):

$$r_i - r_j = y_k \quad (2)$$

where y_k is the margin of victory for game k , and r_i and r_j are the ratings of teams i and j , respectively. In other words, the difference in the ratings r_i and r_j of the two teams

ideally predicts the margin of victory in a contest between these two teams. The Massey method uses point score data to rate teams. However, other game statistic could be used.

3. Related Work

The problem of modeling the outcomes of matches in association football (soccer) has been studied for several years. According to [Goddard 2005], we can divide the previous work in this area into two approaches: goal-based and result-based approaches.

The first approach, defended by most applied statisticians, involves modeling the number of goals scored and conceded directly. The previous works [Maher 1982][Dixon and Coles 1997][Lee 1997][Karlis and Ntzoufras 2003] proposed to predict the goals scored and conceded by each team and to infer the actual winner of the match from this result. In those works, a common approach is the Poisson distribution goal-based data analysis whereby match results are generated by appropriate parameters. Maher (1982) introduced attack and defense parameters for each team and modeled the amount of goals scored by both teams with a Poisson distribution. Nevertheless, the model could not be used to predict scores or results for future matches. In [Dixon and Coles 1997] [Rue and Salvesen 2000], this model was extended, so that it can be used for prediction.

The second approach was developed by a number of applied econometricians and involves modeling the outcome (win-draw-loose) of the game directly, using discrete choice regression models such as ordered probit or logit [Goddard 2005]. An ordered probit regression model was proposed for this purpose in [Goddard and Asimakopoulos 2004]. Koning (2000) proposed this in his research about balance in the Dutch national football league. Goddard (2005) implemented a similar model in a comparative study and concluded that Poisson models and ordered regression models are the most used methods for predicting football. In his comparative research, he concludes also that these methods have a very similar performance.

Models based on team quality ratings also were proposed for football prediction. Leitner et. al. (2010) proposed the prediction of the outcome of the 2008 European championship based on ELO ratings (initially developed for assessing the strength of chess players [Elo 1978]) and bookmaker odds. In [Hvattum and Arntzen 2010], the authors presented a method that used the ELO rating aiming at match predictions and concluded that ratings appeared to be useful for measuring the strength for a team. In [Luckner et al. 2008], it is proposed to predict football outcomes with the help of prediction markets. According to [Constantinou et al 2012], these models based on team quality ratings do not appear to have been extensively evaluated.

Other researchers have proposed techniques of machine learning for prediction. In [Tsakonas et al 2002], the authors compared a genetic programming based technique to other two methods based on fuzzy models and neural networks. Rotshtein et al. (2005) used a genetic algorithm and fuzzy model to predict the outcome of games. Rowan (2007) evolved betting strategies for betting on the outcome of football games, and was able to get a profit with this model. In [Hoekstra 2012], it is introduced a prediction model where an ensemble of k-nn prediction models is created with the help of an evolutionary algorithm. The authors, in [Segundo et al. 2015], have proposed to apply Support Vector Machines (SVM), Decision Trees and Random Forest for predicting the results of soccer matches of the Brazilian Championship A Series.

Bayesian networks and subjective information have also been considered for football match predictions. In [Min et al. 2008], the authors presented a model that combined a Bayesian network along with a rule-based reasoner to provide reasonable World Cup forecasts through simulating various predefined strategies along with subjective information. Constantinou et al. present other works that defend this strategy in [Constantinou et al 2012] and propose a new Bayesian network model for forecasting the outcomes of football matches in which the subjective variables represent the factors that are important for prediction but which historical data fails to capture. In [Owramipur et al. 2013], the authors propose a Bayesian network to predict results of football matches involving Barcelona FC. The authors, in [Joseph et al 2006], present predictions for football results using Bayesian nets and other machine learning techniques.

Although there are models which have got promising results, the problem of predicting football match outcomes is not still well solved. In this paper, we propose a model that uses a Bayesian network to analyze information obtained online about tournaments and to carry out the prediction of outcomes of football matches. Unlike previous works which apply Bayesian networks, our approach also proposes to use the Massey ranking method aiming at to improve the probability of the prediction.

4. Model Description

In this paper, we propose a model that applies a Bayesian network supported by Massey method for predicting the outcomes of soccer matches. The Bayesian network is induced to represent and estimate probabilities in a scenario in which data are stochastic and incomplete. The Massey method provides a margin of victory in a contest between two teams based on the difference in the ratings these two teams ideally. The proposed model aims at to analyze the probability of a team winning, drawing or losing a match.

To validate the proposed prediction model, we use 2014 Brazilian Championship A Series data (commonly referred to as “Campeonato Brasileiro” and popularly as “Brasileirão”), in which the best teams of the country are. Thus, it is worth believing that every team has a chance to win. However, the teams can go through good and bad phases during the season, so that several factors can influence their performance during the championship. Therefore, each team can be at different moments in each game, allowing different predictions of match outcome during the championship. The data are obtained by web-crawler through a sport site in <globoesporte.globo.com>.

The proposed Bayesian network structure is composed by eight nodes: team historic, goal balance, home team, Massey ranking, chances by score, player level, weighted ranking and victory. The choice of these nodes has been inspired by [Min et al. 2008], [Farias 2008] and [Arruda 2000].

historic team: this node represents the probability of a team A winning (Prob(team A)) a team B according to goals number scored (scoredGoal) during every seasons of the “Brasileirão”. The historic team node has its value calculated according to equation (3):

$$\text{Prob}(\text{team A}) = \frac{\text{scoredGoal}(\text{team A})}{\text{scoredGoal}(\text{team A}) + \text{scoredGoal}(\text{team B})} \quad (3)$$

For instance, considering that the teams “Cruzeiro” and “Portuguesa” scored 1975 and 961 goals, respectively, then the probability of “Cruzeiro” winning “Portuguesa” will be 0.67; and, otherwise, 0.32.

goal balance: this node is similar to historic team node, but it represents the probability of a team winning considering its goals scored only during the current season.

home team: the advantage of playing at home is discussed in [Nevill et al. 2002]. The authors point out four reasons that make the home team takes advantage of the visiting team: larger crowd, local referee, familiarity with the field and fatigue of the visiting team due to travel. In [Almeida 2011], the authors have concluded that there is an advantage of around 69% of the home team winning. Thus, in this paper, we have considered 70% as the probability of the home team winning, within the standard deviation range calculated by [Almeida 2011].

Massey ranking: this node represents the probability of a team winning based on rating value calculated by Massey’s ranking method (see section 2.2). The value of the Massey ranking node for each team is normalized in range [0, 1]. For the initial experiments, it was necessary to use the real result of the games of the chosen championship (2014 “Brasileirão”).

chances by score: the “Campeonato Brasileiro” classifies the teams according to scores. Thus, this node represents the probability of a team winning taking into account its score obtained in the competition.

player level: the <globoesporte.globo.com> site provides an individual evaluation of the team players carried out by soccer specialists. Thus, each player can be classified as: “he is selection”, “he plays very much”, “he adds value”, “he composes cast”, “he may be booed” and “he annoys the crowd”. Then, this node receives the sum of the individual classification of the players multiplied by a weight that represents the team classification and it is defined here arbitrarily as: 18, 13, 7, 4, 2 e 1. After that, the values are normalized in range [0, 1].

weighted ranking: this node aims at to weight the ranking of the teams so that top teams do not have excessive advantages over low teams.

victory: this node represents the probability of a team A winning a team B. Thus, it responds the query: “What is the probability of the team A winning team B?”. If the probability is greater than 66%, then it means team A wins; if the probability is between 33% and 66%, then it means draw; and if the probability is smaller than 33%, then it means the team A losses.

For the initial experiments, we have decided constructing the Bayesian network structures manually without a Bayesian network learning algorithm until the structure presenting the best classification results was induced. Fig. 2 presents the Bayesian network induced manually for the proposed prediction model.



Figure 2. Bayesian network induced for the proposed prediction model.

5. Experiments and Analysis of Results

The proposed prediction model in this paper for forecasting outcomes football matches was validated using 2014 Brazilian Championship A Series data, also known as “Brasileirão”.

The predictions have been accomplished using the joint probability distribution represented through the Bayesian network of Fig. 2, according to the equation (1). Thus,

$$P(\text{historic team, goal balance, \dots, victory}) = P(\text{victory} \mid \text{historic team, goal balance, home team, weighted ranking, player level}) * P(\text{weighted ranking} \mid \text{Massey ranking, chances by score}) * P(\text{historic team}) * P(\text{goal balance}) * P(\text{home team}) * P(\text{Massey ranking}) * P(\text{chances by score}) * P(\text{player level})$$

Considering that the chosen championship has 38 rounds, where 20 teams play against each other in each round, initially it was analyzed the inference quality of the proposed prediction model in each match. In a total of 380 matches during all season, our model has got the best results in the prediction of the victory number. In 197 matches where a team A has won a team B, the proposed prediction model has correctly inferred 90. Already in 92 matches where there have been draws, the proposed prediction model has correctly predicted 31 only. The number of losses predicted correctly is 37 in 91.

In some rounds, the model has correctly inferred the outcome of 7 matches in 10; in others, it has correctly inferred the outcome of 1 match in 10 only. Table 1 presents the number of matches correctly inferred in each round by proposed prediction model. It was identified that the model has correctly inferred 40.78% of the match outcomes of the “2014 Brasileirão”. This result is comparable to accuracy of the learners presented in [Joseph et al. 2006], which are: decision trees (30.19% and 38.65%), Naive Bayes (38.81% and 35.74%), Data Driven Bayesian learner (32.31% and 32.62%) and K-nearest neighbor (KNN) (34.98% and 37.06%).

For validation of the proposed prediction model, we have also analyzed the value of the F1-Score metric in every round of “2014 Brasileirão”, given by the equation (4) [Bilenko et al. 2003]:

Table 1. Number of matches correctly inferred in each round of the “2014 Brasileirão” by proposed prediction model.

Number of matches correctly inferred	Round
1	13, 36 and 37
2	21, 30 and 35
3	7, 9, 14, 15, 19, 24, 25 and 31
4	8, 16, 17, 18, 28 and 33
5	3, 5, 10, 12, 22, 27, 32, and 38
6	1, 2, 11, 20, 26 and 29
7	4, 6, 23, 24, and 34

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

The Recall presents a measure of completeness. It is calculated as the fraction of correct instances among all instances that belong to relevant subset and is given by equation (5):

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (5)$$

The Precision is considered as an accuracy measurement, calculated as the fraction of correct instances among those ones that the algorithm judged to belong to relevant subset and it is given by the equation (6):

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (6)$$

Since the event of a football match is tricotomic (win-draw-loss), it was decided evaluating the inference as follow: “true positive” receives 1 point when the model hitting the prediction of victory; “true negative” receives 1 point when the model hitting the prediction of loss; “false positive” receives 1 point when the model infers the victory, but the real result is loss and it receives a proportional value in range [0, 1], according to equation (7), when the model infers the victory but the real result is draw; “false negative” receives 1 point when the model infers the loss, but the real result is the victory and it receives a proportional value in range [0, 1], according to equation (7), when the model infers the loss but the real result is draw.

$$Proportional\ Value = 1 - \left(\frac{\left(\frac{ProbOfVictory}{0.3333} - 100 \right)}{100} \right) \quad (7)$$

Thus, every round of the “2014 Brasileirão” has been evaluated using true positive cases and F1-Score. The proposed prediction model has presented arithmetic mean of 70.07% for F1-Score metric with standard deviation of 0.1755. Table 2 presents every round and the values for true positive, false positive and false negative, Precision, Recall and F1-Score.

Table 2. Prediction results of true positive (TP), false positive (FP), false negative (FN), Precision, Recall and F1-Score obtained by proposed model in each round of the “2014 Brasileirão”.

Round	TP	FP	FN	Precision	Recall	F1-Score
1	6	1.00	0.00	0.86	1.00	0.92
2	5	1.83	1.00	0.73	0.83	0.78
3	5	1.92	1.00	0.72	0.83	0.77
4	7	2.86	0.00	0.71	1.00	0.83
5	5	0.00	2.18	1.00	0.70	0.82
6	7	0.30	0.90	0.96	0.89	0.92
7	3	1.76	2.09	0.63	0.59	0.61
8	4	0.41	2.00	0.91	0.67	0.77
9	3	3.25	0.00	0.48	1.00	0.65
10	5	1.29	0.54	0.79	0.90	0.85
11	6	1.72	1.00	0.78	0.86	0.82
12	5	0.07	3.19	0.99	0.61	0.75
13	1	3.81	0.00	0.21	1.00	0.34
14	3	3.55	1.00	0.46	0.75	0.57
15	3	1.00	2.00	0.75	0.60	0.67
16	4	4.63	1.00	0.46	0.80	0.59
17	4	1.50	1.89	0.73	0.68	0.70
18	4	1.14	0.03	0.78	0.99	0.87
19	3	1.01	1.00	0.75	0.75	0.75
20	6	1.05	1.00	0.85	0.86	0.85
21	2	6.04	0.00	0.25	1.00	0.40
22	5	1.83	1.00	0.73	0.83	0.78
23	7	0.34	1.00	0.95	0.88	0.91
24	3	4.64	0.00	0.39	1.00	0.56
25	3	2.63	1.00	0.53	0.75	0.62
26	6	2.71	1.00	0.69	0.86	0.76
27	5	2.40	0.30	0.68	0.94	0.79
28	4	4.00	2.00	0.50	0.67	0.57
29	6	1.00	0.56	0.86	0.91	0.88
30	2	1.62	1.98	0.55	0.50	0.53
31	3	3.38	0.00	0.47	1.00	0.64
32	5	0.75	1.89	0.87	0.73	0.79
33	4	1.52	1.74	0.72	0.70	0.71
34	7	0.24	0.73	0.97	0.91	0.94
35	2	0.26	4.19	0.89	0.32	0.47
36	1	4.37	0.89	0.19	0.53	0.28
37	1	3.55	1.00	0.22	0.50	0.31
38	5	1.17	0.41	0.81	0.92	0.86

Although the Bayesian network constructed manually has considered several characteristics capable of influencing the outcome of the game, it has not yet been able to absorb the highly stochastic nature of the soccer. For example, in some rounds, the team more probable of winning used reservation players and it lost the matches while the proposed prediction model had predicted the victory. However, the proposed prediction model has been satisfactory when by analyzing all matches of a round. In round 1, for instance, it has correctly inferred 92% of the outcomes, according to F1-score metric (see Table 2). Therefore, it is worth pursuing the proposed model for prediction of soccer outcomes.

6. Conclusion and Future Work

In this paper, we present a prediction model for forecasting soccer match outcomes. The proposed prediction model applies Bayesian networks and Massey ranking method for predicting the outcome (win-draw-loose) of the game directly. Massey ranking method provides the probability of a team winning according to a rating ordered vector representing the team rank.

The proposed prediction model has been validated using data extracted from the site <http://globoesporte.globo.com/> through web-crawler. For the initial experiments, we used 2014 Brazilian Championship A Series data, popularly known as “Brasileirão”.

Considering the highly stochastic nature of the soccer matches, the obtained results have been satisfactory. Although the proposed prediction model has not correctly inferred some match outcomes, it has obtained the arithmetic mean of 70.07% for F1-Score metric of correct predictions in the rounds of championship. Thus, it is believed that the proposed prediction model is promising.

Based on results, we intend next to augment the database and investigate other attributes to be inserted in the Bayesian network. Taking into account the availability of online platforms such as twitter, it is possible to explore the massive amounts of collective knowledge and use them for prediction. Besides, we will perform a study on learning and inference Bayesian algorithms to improve the performance of the proposed prediction model.

References

- Almeida, L. G. de, Oliveira, M. L. de, Silva, C. D. da. (2011). Uma análise da vantagem de jogar em casa nas duas principais divisões do futebol profissional brasileiro. *Revista Brasileira de Educação Física e Esporte*, 25, (1), 49–54.
- Arruda, M. L. de. (2000). *Poisson, Bayes, Futebol e DeFinetti*. Master’s thesis. University of São Paulo, Brazil.
- Bilenko, M., Mooney, R., Cohen, W., Avikumar, P., Fienberg, S. (2003). Adaptive name matching in information integration. *IEEE Intelligent Systems, IEEE*, 5, 16–23.
- Constantinou, A. C., Fenton, N. E., and Neil, M. (2012). pi-football: A bayesian network model for forecasting association football match outcomes. *Knowledge-Based Systems*, 36:322–339.

- Crowder, M., Dixon, M., Ledford, A., Robinson, M. (2002). Dynamic modelling and prediction of English football league matches for betting. *The Statistician*, 51, (2002), 157–168.
- Dixon, M. J., & Coles, S. C. (1997). Modeling association football scores and inefficiencies in the football betting market. *The American Statistician*, 46, 265–280.
- Elo, A.E. (1978). *The Rating of Chess Players, Past and Present*. Arco Publishing, New York.
- Farias, F. F. (2008). *Análise e previsão de resultados de partidas de futebol*. Master's thesis. Federal University of Rio de Janeiro, Brazil.
- Goddard, J. and Asimakopoulos, I. (2004). Forecasting football results and the efficiency of fixed-odds betting. *Journal of Forecasting*, 23, 51–66.
- Goddard, J. (2005). Regression models for forecasting goals and match results in association football. *International Journal of Forecasting*, 21, 331–340.
- Hoekstra, V. (2012). *Predicting football results with an evolutionary ensemble classifier*. Master's thesis, VU University Amsterdam, October.
- Heckerman, D., Geiger, D., Chickering, D. (1995). Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20 (3), 197-243.
- Hvattum, L. M., Arntzen, H. (2010). Using ELO ratings for match result prediction in association football. *International Journal of Forecasting*, 26, 460–470.
- Joseph, A, Fenton, N. E., Neil, M. (2006). Predicting football results using Bayesian nets and other machine learning techniques. *Knowledge-Based Systems*, 19, (2006), 544–553.
- Karlis, D., Ntzoufras, I. (2003). Analysis of sports data by using bivariate Poisson models. *The Statistician*, 52 (3), (2003), 381-393.
- Koning, R. H. (2000). Balance in competition in dutch soccer. *Journal of the Royal Statistical Society. Series D (The Statistician)*, (3).
- Langville, A. N. (2012). *Who's #1?: The science of rating and ranking*. Princeton University Press, New Jersey.
- Lee, A. J. (1997). Modeling scores in the Premier League: is Manchester United really the best?, *Chance* 10, (1997), 15-19.
- Leitner, C., Zeileis, A. and Hornik, K. (2010). Forecasting sports tournaments by ratings of (prob)abilities: A comparison for the euro 2008. *International Journal of Forecasting*, 26(3):471 – 481.
- Luckner, S., Schroder, J. and Slamka, C. (2008). On the forecast accuracy of sports prediction markets. In *Negotiation, Auctions, and Market Engineering, volume 2 of Lecture Notes in Business Information Processing*, 227–234. Springer Berlin Heidelberg.
- Maher, M. J. (1982). Modeling association football scores. *Statistical Neerlandica*, 36, 109-118.

- Min, B., Kim, J., Choe, C., Eom, H., McKay, R.B. (2008). A compound framework for sports results prediction: a football case study. *Knowledge-Based Systems*, 21, (2008), 551–562.
- Neapolitan, R. E. (2003). *Learning Bayesian networks*. New Jersey: Prentice Hall, Inc.
- Nevill, A. M., Balmer, N. J., Williams, A. M. (2002). The influence of crowd noise and experience upon refereeing decisions in football. *Psychology of Sport and Exercise*, Elsevier, 3 (4), 261–272.
- Owramipur, F., Eskandarian, P., Mozneb, F. S. (2013). Football result prediction with Bayesian network in Spanish league-Barcelona team. *International Journal of Computer Theory and Engineering*, 5, (5), 812-815.
- Pearl, J. 1988. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann.
- Rotshtein, A., Posner, M. and Rakityanskaya, A. (2005). Football predictions based on a fuzzy model with genetic and neural tuning. *Cybernetics and Systems Analysis*, 41:619–630.
- Rowan, M. (2007). *Evolving strategies for prediction of sporting fixtures*. Master's thesis, University of Birmingham.
- Rue, H. and Salvesen, O. (2000). Prediction and retrospective analysis of soccer matches in a league. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49(3), (2000), 399-418.
- Segundo, C. A. B., de Abreu, A. A. M., Esmín, A. A. A. (2015). Previsão de resultados de jogos do campeonato brasileiro de futebol: uma abordagem de mineração de dados. In: *Brazilian Conference on Intelligent System (BRACIS) and Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, Natal, Brazil.
- Tsakonas, A., Dounias, G., Shtovba, S., Vivdyuk, V. (2002). Soft computing-based result prediction of football games. In: *The First International Conference on Inductive Modelling (ICIM'2002)*, Lviv, Ukraine.

Classification of Microcalcifications in Mammographic Exams: optimization with the use of hybrid Artificial Neural Network-Genetic Algorithm (GA-ANN)

Dario Brito Calçada¹, Cláudio Fabiano Motta Toledo¹, Pedro Victor Carvalho das Chagas²,
Cássio Costa Almeida²

¹Instituto de Ciências Matemáticas e de Computação
USP - Universidade de São Paulo. São Carlos, Brasil

²Universidade Estadual do Piauí, UESPI
Parnaíba, PI, Brazil

dariobcalcada@usp.br, claudio@icmc.usp.br, {pedrocarvalhophb, jcassiocosta}@gmail.com

Abstract. *Breast cancer is the most severe among women, and mammography is the most effective screening test for microcalcifications that may lead to the onset of this disease. Breast examination, through mammography, is extremely sensitive to the professional's interpretation. Artificial Intelligence (AI) stands out as an innovative tool that allows a better analysis, the use of Artificial Neural Networks and Genetic Algorithms has been expanding in the areas of health care systems. This work is based on Digital Image Processing techniques to perform the extraction of morphological information from calcifications and the application of Artificial Neural Network (ANN) and Genetic Algorithm (GA) to classify the calcifications analyzed, in addition to performing a study of parameters involved in mammograms in order. With the use of a hybrid system with Genetic Algorithm evolving the ANNs, the success indexes were expanded, allowing the future generation of a more efficient breast cancer diagnosis system.*

1. Introduction

Breast cancer is the most common cancer among women. According to the National Cancer Institute (INCA), the estimate is that 22% of new cases arise each year. In Brazil, the data computed by the Ministry of Health are 57 cases per 100,000 women. The number of death records resulting from breast cancer continues elevated, mainly because the diagnosis of the disease is made at an advanced stage [INCA 2016].

The most effective way to detect breast cancer is mammography. The examination is made from a device called a mammography, being performed by positioning the breast on the equipment, on a type of double acrylic plate. The main characteristic of mammography is the ability to detect microcalcifications from 1 millimeter (mm) (Figure 1), increasing the chances of cure by up to 95% [Fogaça and Garrote 2004].

Mammography is considered the most reliable for the screening of breast cancer due to its Sensitivity. However, the interpretation of lesions on a mammogram is a complex task for specialists, whose experience depends on an accurate diagnosis. Distortions

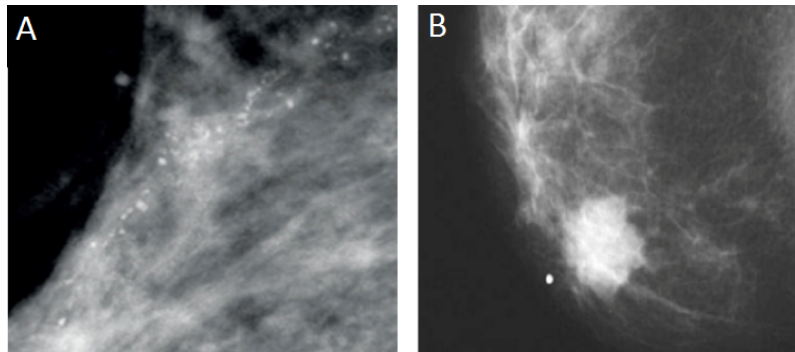


Figure 1. Calcification clusters (A) and irregularly shaped Nodules (B) (Digital Database for Screening Mammography (DDSM) University of South Florida research program).

in the interpretation and classification of suspected lesions by specialists imply a larger number of unnecessary biopsies. It is estimated that between 65% and 85% of breast biopsies are performed in benign lesions [Stravos 2009].

In the last two decades several works using techniques of Image Processing, Pattern Recognition, Computer Vision and Artificial Intelligence have been developed as part of Computer Aided Diagnosis (CAD) schemes, which can function as a second opinion to radiologists as to the location and nature of these structures [Drukker et al. 2010].

Specialists classify the exams results in a standard system called BI-RADS (Breast Imaging Reporting and Data System) that was developed in 1993 to minimize the divergences in the interpretations of the images and to facilitate the control of results [ACR 2005]. This system classifies the results on a scale ranging from I to V, where I represents small risk and V large risk. The diagnosis made by the specialists, through mammography exams based on BI-RADS, is very subjective since it depends on the quality of the images. In these diagnoses, suspicious areas are previously identified as ROI (Region of Interest) that characterize the interest areas of the images.

The problem of the diagnoses generates other disturbances in the patients, such as depression, which causes greater difficulty in the treatments [Ho et al. 2015]. To aid the diagnosis previous of the cancer and the psychological symptoms related to it, one can make use of techniques of Intelligence Computational [Cvetkovic and Cvetkovic 2017] such as Artificial Neural Networks and Genetic Algorithms [Gohariyan et al. 2017]. Therefore, it is extremely important to identify methods that drastically reduce the number of unnecessary biopsies, as well as to identify the main characteristic that assists specialists in the correct diagnosis of microcalcification.

2. Overview: Problem and Methodology

The present work seeks to deal with problems related to the complexity and subjectivity involved in the process of determining the diagnosis of breast cancer, when related to the image quality and the morphological characteristics of the patient. These factors can lead to false positive results in clinical examinations and, consequently, unnecessary biopsies. For example, visual changes from external factors during mammography may alter the parenchyma of reading images [Louveira et al. 2003].

This work is based on techniques of Digital Image Processing (DIP) to perform the extraction of morphological information from calcifications, in which computational intelligence techniques such as Artificial Neural Networks Multilayer Perceptron (ANN MLP) and Genetic Algorithm (GA) are applied to classify the calcifications analyzed according to the morphological classification defined by BI-RADS. ANNs are able to perform the classification of patterns by learning through examples [Jasmine 2009]. In addition to the ANNs that are inserted within the area of intelligent systems (connectionist) or computational intelligence [Silva et al. 2010], we can mention other methods such as Genetic Algorithms and Fuzzy Logics, besides approaches based on the fusion of these techniques, i.e., the famous hybrid algorithms [Shapiro 2002]. Genetic Algorithms (GAs) are computational intelligence techniques that are part of the group of evolutionary algorithms, methods inspired by the biological concept of the natural evolution of the species [Eiben and Smith 2007].

The performance of ANN MLP in this work was evaluated in a learning and validation test with real calcifications extracted from mammograms from Digital Database for Screening Mammography (DDSM). An analysis of ANN MLP parameters, aiming to improve classification, was also performed by applying the decision tree algorithm J48. After the study of the parameters, we assembled a new ANN and an GA capable of developing a better resolution of the proposed problem.

The research used the DDSM, which has a wide range of digitized mammograms and classified images. The purpose of the work was to follow the steps:

- Characterize the points of the images needed to extract the parameters;
- Identify the most efficient method for treatment of scanned images;
- Adjust the relevant parameters for the ANN;
- Characterize the positives and negatives of the tests.
- Make the study of the parameters in order to obtain the one that has the greatest influence on the classification BI-RADS
- Apply hybrid algorithm combining ANN and GA to obtain a high accuracy classifier

The steps mentioned are detailed in the following sections.

3. ANN data acquisition and adjustment

The acquisition of the data was performed from the images acquired from the digital image repository developed by the University of South Florida, which has in its collection a wide range of digitized and properly characterized images (DDSM). Provides a frame structure of images, standardized in the size of 230 x 350, discriminating the characteristics of margins, contours and BI-RADS classification of each exam, identifying in the images the points of interest (ROI).

The selected parameters of the available reports were the margins, shape, assessment, age and density. These parameters are listed in a file of type ".txt".

In the case of the shape and margin parameters, a scale was established to quantify the data according to the degree of the same attributes assigned to BI-RADS [Kopans 2000]. Table 1 presents this scale.

Table 1. Quantification of methods of form and margin.

Shape	Margin	Value
Round	Circumscribed	1
Ovulate	Blurred	2
Lobular	Microlobulated	3
Irregular	Poorly Defined	4
Architectural Distortion	Spiked	5

The characterization of the extracted ROI was carried out by the construction of a *script* using the MATLAB® tool. This script served to identify the edges found and extract the size (cmsup 2) of the ROIs in each image. The images were standardized in the dimensions 230 x 350, as established by the DDSM.

At the end of the ROIs identifications procedure for individual calcifications delimiting the area of interest, in total 150 samples were distributed among the 5 classification possibilities analyzed in the scope of this work, According to the nomenclature and the criteria of differentiation of the BI-RADS [ACR 2005].

MATLAB® software, version R2013a (8.1.0.604), was used to create, train, validate and test the ANNs. The ANN defined for the morphological classification [ACR 2005] of the ROIs consists of the function of the MLP model with the learning algorithm *backpropagation*. In total 5 (five) input parameters were used by ANN that correspond to the quantity of morphological information extracted from ROIs. The exception was the BI-RADS parameter, applied only in the analysis of calcification results.

The elements were separated into three groups in order to obtain results for each type of classification and to verify the efficiency of the distribution of the elements in ANN as listed in Table 2. A maximum limit of 200 times was used for training and 15 cross-validations for each topology of the network, and for each validation 10 training tests were performed. For each step, the respective indices hit were calculated in order to compare the performance of the algorithms and to obtain the ANN with the lowest mean square error.

In the training stages, the weights of the connections between pairs of neurons can undergo alterations, which characterizes the learning aspect of the ANN algorithm. In the validation step, the weights can or can not be modified, comparing with other ANNs of the same topology in order to always obtain the best result. At the end of these steps, ANN is used with previously unused data to calculate its accuracy in relation to the reception of new information, in this step the weights are no longer altered. Due to these conditions, it is necessary to calculate the error rates and correctness in each step.

Table 2. Distribution of the data in each ANN stage.

Phase	Number of Experiments	%
Training	90	60.00
Validation	30	20.00
Testing	30	20.00
Total	150	100.00

After the training, validation and testing of ANN, a study of the parameters through the application of the decision tree algorithm was conducted. In this study, we used the WEKA tool (Waikato Environment for Knowledge Analysis) through the decision algorithm J48 ('decision tree C4.5'). By means of this technique it was possible to measure which parameter is most likely to be relevant within the studied set, i.e., to identify which parameter stands out most in the decision tree. With the definition of the most relevant parameter, it was added in duplicate for a new ANN training following the same procedures of the previous step.

In both stages of training the ANNs were varied the number of neurons of the hidden layer. The maximum number of neurons in the first hidden layer followed Kolmogorov's Theorem [Kolmogorov 1957], which states that the number of neurons in the hidden layers must follow Equation $2E + 1$, where E represents on total number of ANN entries. For the second layer the maximum number of neurons was half the value measured at the first. Thus, since the first ANN has 5 entries, the variation of 2-11 (two to eleven) neurons for the first hidden layer and 0-5 (zero to five) neurons for hidden second layer.

For the second ANN, with the addition of one entry, the number of neurons in the first hidden layer ranged from 2-13 (two to thirteen) and the second hidden layer from 0-6 (zero to six). ANNs with only 1 (one) neuron in the hidden layer was not used, because it only generates positive results in classification tasks with linearly separable data, which can not be defined in the domain of the information used.

With the results of the ANNs, we chose to make a hybrid approach with the use of GA, since the improvement of results has already been obtained in other works [Aličković and Subasi 2015], but without the use of the parameter studies.

4. Hybrid system (GA-ANN)

The proposed hybrid system combines ANN with GA to define a network topology that leads to greater accuracy in the results. The implementation of this hybrid system was performed in Java v. 1.8, in which the ANN adopted for classification was the MLP model following the same configuration of inputs and outputs defined in section 3.

For the evolution of the ANNs, a GA with vector chromosomes of the size defined by Equation 1 was implemented. Each entry in this chromosome defines a weight of ANN that should initially be a contained value in the range [-10.0; 10.0].

$$W_n = (I_n * H_n) + (H_n * O_n) + H_n + O_n \quad (1)$$

W_n represents the total number of genes on the chromosome (total ANN weights), I_n the number of ANN input parameters, H_n the number of neurons in the hidden layer of ANN and O_n the amount of neurons in the layer about to leave. The fitness value of the subjects of the population was obtained by means of the calculation of the average quadratic error, in which the individual with the best aptitude is the one with the lowest fitness value. Figure 2 shows a diagram of the main components of the GA used.

A population of 20 individuals is generated with randomly selected weights. Selection for reproduction uses the tournament strategy among 3 chromosomes, randomly

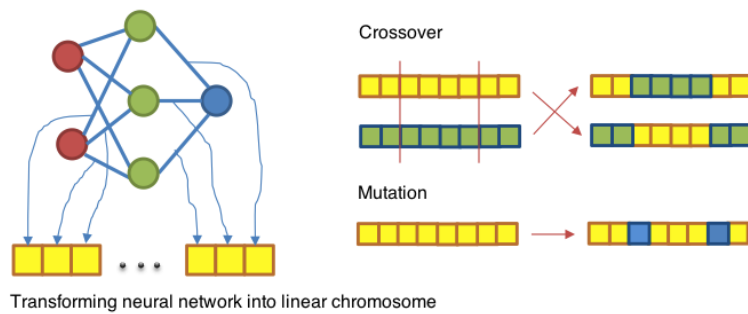


Figure 2. ANN transformation into a linear chromosome [Lahodiuk 2014]

selected from the population, in which the one with the best fitness value is chosen to reproduce. After the selection of two country chromosomes by the tournament, the crossover operator is executed to create two new chromosomes. The crossover with a cut-off point was applied, in which the points are defined in a random way. Selection for survival uses an intermediate population (Offspring), in which the children are inserted. We then select the 20 (twenty) best individuals between the current population (parents) and offspring (children). The mutation operator is then applied with a rate of 10% changing the values in the genes of each new chromosome created. In the composition of the new population at each interaction, a new chromosome created randomly and inserted in the place of the worst individual of the population. The introduction of this new individual seeks to maintain the diversity of the population [McCaffrey 2014].

- Grupo A - 150 samples with single inputs.
- Grupo B - 150 samples with the duplicate density parameter.

The same dynamics of neuron variations that were used in the elaboration of the ANNs were followed for each group, respectively, in which each topology formed a population of the GA and, at the end of the processing, the better topology, as well as the individual (ANN) with less fitness.

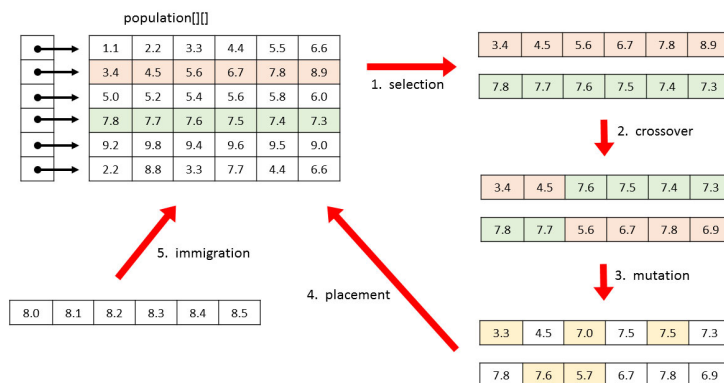


Figure 3. Genetic operators in the hybrid system GA-ANN [McCaffrey 2014]

5. Results and discussion

5.1. Artificial Neural Network

The ANN detected with the highest hit index, that is, the lowest mean quadrupole error has an 8:3:1 topology from the tests performed. This means that there are eight neurons in the first hidden layer, three in the second hidden layer and one neuron in the output layer.

In order to evaluate the results, two different metrics were used: the correlation coefficient (r value) and the percentage absolute error (MAPE) in Equation 2. The first one indicates the intensity and direction is a linear relationship between two random variables. Therefore, it is evaluated how the variations of the output data are explained by the neural model. The second measure indicates how close the predicted results are to the actual data.

$$MAPE = \frac{\sum_{k=1}^n \left| \frac{Y_d - Y}{Y_d} \right|}{n} * 100 \quad (2)$$

In Equation 2, Y_d and Y are the outputs desired and calculated by ANN, respectively. There was a success rate of 91%. Figure 4 shows a dashed linear line corresponding to the ideal condition to be achieved. The line parallel to the dashed line shows how close the result is to the ideal condition. By means of the analysis of parameters identified in the training phase, it was observed that the influence on the hit rate was directly linked to the quantity of samples, since the more samples were destined to this stage, the more chances the neural network would find a good result.

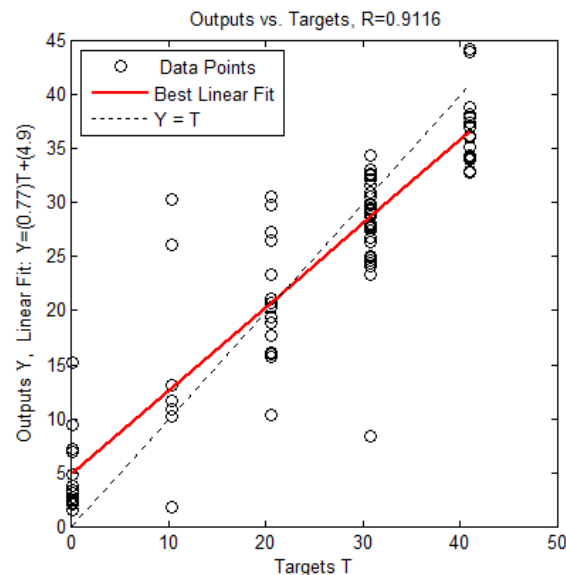


Figure 4. Results of neural network training

The training of the neural network, whose objective was the convergence of the linear function capable of generalizing the input classifications (size, density, shape, age and margin) of the network was successful for the BI-RADS classification.

The values that relate to the training obtained a better correlation result in relation to the training, validation and test values, being 0.9116, 0.8191, 0.5407 as shown in Figure 5. The values above 0.9 are considered excellent for training in various exit possibilities [Vilela Neto 2009].

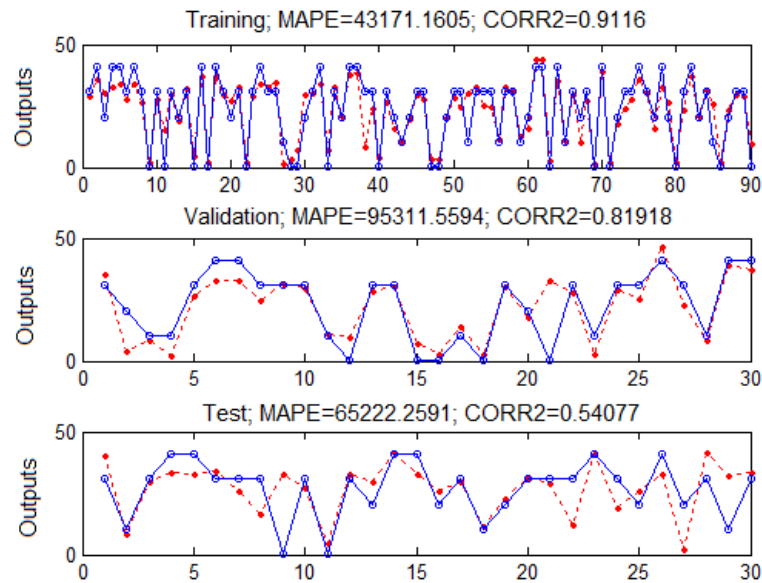


Figure 5. Results of training, validation and tests of the Artificial Neural Network

5.2. Studies of parameters and ANN

The application of the data mining technique through the decision tree made it possible to identify density as the parameter of greatest probability of importance. This finding was possible because this element, besides being the root of the tree, is also not found as in the leaf (Figure 6). This demonstrates the importance of the same to the classification of microcalcification. In this way, it was decided to duplicate this entry for the formation of a new ANN.

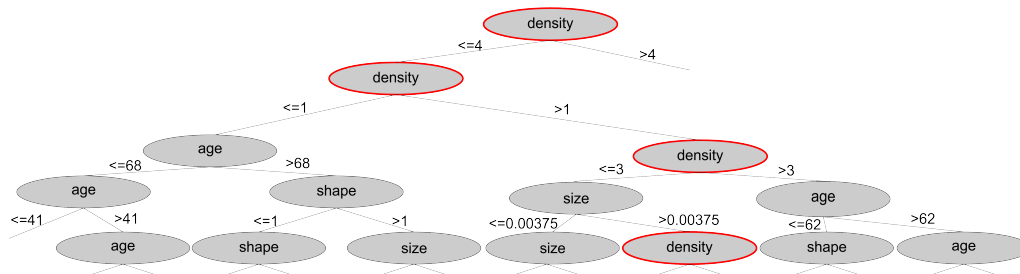


Figure 6. Highlight of the decision tree indicating the density parameter as in the root.

Once the most relevant parameter was determined, a new ANN was constructed in which the density was inserted in duplicate (two entries) in order to obtain optimized results in relation to the first ANN used.

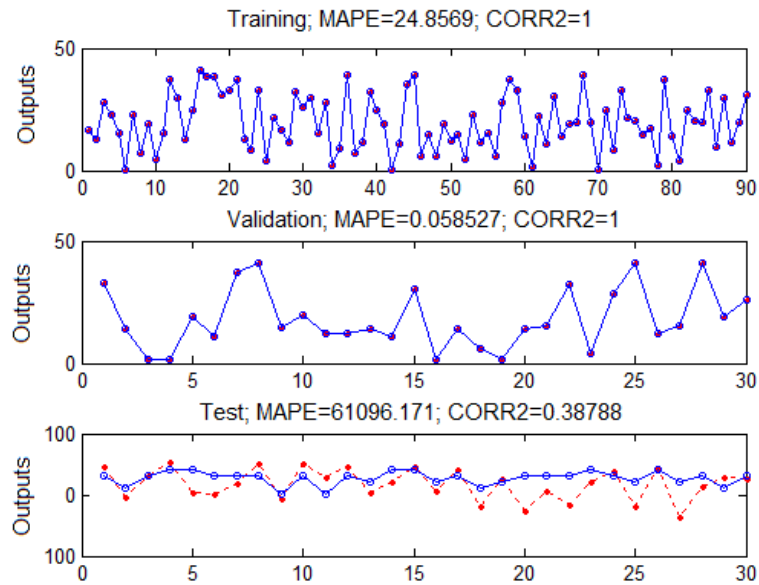


Figure 7. Results of training, validation and tests of the Artificial Neural Network

By means of the tests performed, the ANN that had the highest index of success, that is, obtained the lowest quadrupole error, has a 3:1 topology. Thus, a topology with three neurons in the first hidden layer and a neuron in the output layer, is therefore different from the previous result with an alteration in the number of neurons and in the number of hidden layers.

In this training phase, a 100% hit rate was obtained. When this occurs, caution should be exercised over the possibility of *overfitting* occurring. Overfitting occurs when a machine learning model captures data noise, i.e., occurs when the algorithm adjusts the data very well [Liu and Gillies 2016]. Overfitting can be avoided with the use of several models and cross-validation to effect comparison according to the predictive accuracy in test data. In this work, the probability of having overfitting is minimal, since the data were different from training and validation data.

The values that relate to the hit rates obtained better correlation results in relation to the values of training, validation and testing, being 1, 1, 0.38788, respectively (Figure 7). It can be seen that, in addition to the training improvement, there was also a growth in the test rate when compared to ANN with only single input data, although the test values were not as satisfactory in comparison with other experiments. This can be explained by the fact that they are data related to health and that they have real diagnostic results. Thus, the obtained values can demonstrate an efficiency of generalization of the ANN provoking a test hit of 38%.

5.3. Hybrid Algorithm GA-ANN

In this experiment, the best topology for ANN obtained until then was reevaluated as well as the values of the weights. GA is performed considering variances in the number of neurons in the hidden layer. Thus, the AG is initially run for X neurons in the first hidden layer up to a total of Y neurons. As the quadratic error medium was used, at each GA

generation the indexes became smaller, until the convergence of the values occurred and the change would not become but significant, causing the stop of the evolutionary algorithm. If this did not occur until the generation of $200 \cdot 10^6$, the algorithm was terminated for that topology and restarted with the next, as was done initially with the ANNs. Each set of experiments was individually analyzed as shown in Figures 8 and 9.

When comparing the figures of the best ANNs found for each group, it is noticed that the insertion of the density parameter caused a later conversion, varying from 50 generations to 80 generations. However, when we checked the individuals, we noticed that the fitness of group B has undergone a greater refinement, and, consequently, obtaining better values.

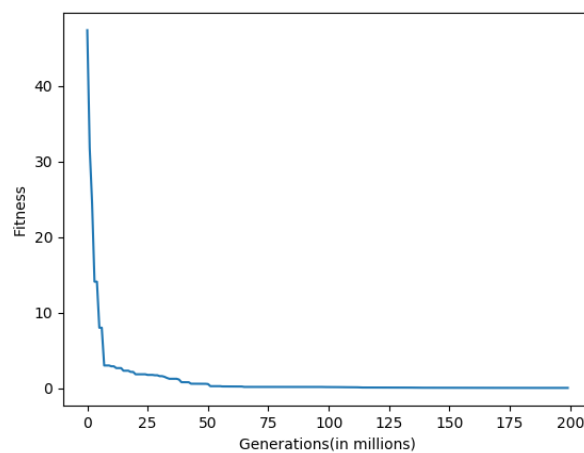


Figure 8. Fitness x Generations (Group A)

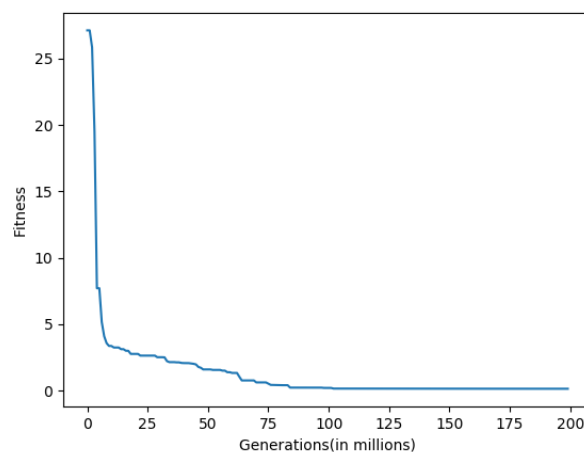


Figure 9. Fitness x Generations (Group B)

For group A, the best topology at the end of the evolution process was 7:1, i.e., 7 in the first hidden layer and 1 neuron in the exit layer. The gain in the increment of an evolutionary approach made possible an index of success of 96% in testing phase,

this index indicating a possibility of direct use of the technology, since it optimizes the accuracy of the method in test phases.

The best topology found for group B was also 7:1, that is, 7 in the first hidden layer and 1 neuron in the output layer, but with a success rate of 97.33%. With this result, the importance of the density parameter for the BI-RADS diagnosis of mammograms is confirmed, as well as the improvement in the accuracy with the duplication of the same.

6. Conclusion

The application of Artificial Intelligence to problem solving is once again found to be important, but it is important that the methodology chosen and its implementation be well reshaped so that the results are optimized.

The use of AG for the evolution of ANNs is an efficient way of obtaining the optimized results with respect to processing time, with a considerable gain in relation to the process Cross-validation of the *backpropagation* algorithm.

The hybrid system obtained a hit rate of 97.33%, which makes the use of this type of technique promising for the development of diagnostic aids to minimize medical errors promising.

For future work, the application of other techniques, such as the use of particle swarm optimization for ANN training, and Genetically Optimized Neural Network algorithms [Bhardwaj and Tiwari 2015] for mammography classification, as well as the use of actual data obtained directly from breast cancer treatment and diagnosis clinics for further practical applications and the development of a diagnostic aids system.

Acknowledgment

Financial support was received from the CAPES, Brazil.

References

- ACR (2005). *BI-RADS: Sistema de laudos e registro de dados de imagem da mama*. American College of Radiology, São Paulo, 1st edition.
- Aličković, E. and Subasi, A. (2015). Breast cancer diagnosis using GA feature selection and Rotation Forest. *Neural Computing and Applications*, 28:753–763.
- Bhardwaj, A. and Tiwari, A. (2015). Breast cancer diagnosis using Genetically Optimized Neural Network model. *Expert Systems with Applications*, 42(10):4611–4620.
- Cvetkovic, J. and Cvetkovic, M. (2017). Investigation of the depression in breast cancer patients by computational intelligence technique. *Computers in Human Behavior*, 68:228–231.
- Drukker, K., Pesce, L. L., and Giger, M. L. (2010). Repeatability and classifier in computer-aided diagnosis for breast ultrasound. *Medical Physics*, 37(6):2659–2669.
- Eiben, A. E. and Smith, J. E. (2007). *Introduction to Evolutionary Computing*. Springer, 2nd edition.
- Fogaça, E. I. C. and Garrote, L. F. (2004). Câncer de mama: atenção primária e detecção precoce. *Arquivos de Ciência da Saúde, São José do Rio Preto*, 11(3):179–181.

- Gohariyan, E., Esmailpour, M., and Mehdi Shirmohammadi, M. (2017). The Combination of Mammography and MRI for Diagnosing Breast Cancer Using Fuzzy NN and SVM. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(5):20–24.
- Ho, S. Y., Rohan, K. J., Parent, J., Tager, F. A., and McKinley, P. S. (2015). A longitudinal study of depression, fatigue, and sleep disturbances as a symptom cluster in women with breast cancer. *Journal of Pain and Symptom Management*, 49(4):707–715.
- INCA (2016). Instituto Nacional do Câncer.
- Jasmine, J. S. L. (2009). Microcalcification detection in digital mammograms based on wavelet analysis and neural networks. In *Control, Automation, Communication and Energy Conservation, INCAEC 2009*, Perundurai, Tamilnadu.
- Kolmogorov, A. N. (1957). On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk*, 114:679–681.
- Kopans, D. (2000). *Imagem da Mama*. MEDSI, Rio de Janeiro.
- Lahodiuk, Y. (2014). Evolving of neural networks for collective agents.
- Liu, R. and Gillies, D. F. (2016). Overfitting in linear feature extraction for classification of high-dimensional image data. *Pattern Recognition*, 53:73–86.
- Louveira, M. H., Souza, L. R. M. F., Castro, I. M., Elias, S., Szejnfeld, J., and Kemp, C. (2003). Avaliação e classificação ultra-sonografia dos nódulos mamários: revisão da literatura e aspectos atuais. *Revista Imagem*, 23(2):123–128.
- McCaffrey, J. (2014). *Neural Network How-To: Code an Evolutionary Optimization Solution*.
- Shapiro, A. F. (2002). The merging of neural networks, fuzzy logic, and genetic algorithms. *Insurance: Mathematics and Economics*, 31(1):115–131.
- Silva, I. N., Spatti, D. H., and Flauzino, R. A. (2010). *Redes Neurais Artificiais: para engenharia e ciências aplicadas*. Artiliber.
- Stravos, A. T. (2009). New advances in breast ultrasound: computer-aided detection. *Ultrasound Clinics*, 4(3):285–290.
- Vilela Neto, O. P. (2009). *Projeto, otimização, simulação e predição de propriedades de nanoestruturas através de técnicas da inteligência computacional : nanotecnologia computacional inteligente*. PhD thesis, PUC-RJ.

A quantitative comparison of exhaustive discriminative pattern mining algorithms

Maurício A. S. dos Santos¹, Renato Vimieiro¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50732-970 – Recife – PE – Brasil

{mass2, rv2}@cin.ufpe.br

***Abstract.** This paper aims at comparing quantitatively the behavior of exhaustive approaches of Discriminative Patterns (DPs) in high dimensional data. Since most algorithms for DP mining have been proposed aiming traditional low dimensional data, such a comparison is critically needed to investigate whether these approaches could be also applied to high dimensional data. We conducted quantitative experiments for the evaluation of the performance of the approaches. The results demonstrate the algorithms are not the most appropriate of the high dimensionality data, however two algorithms are better on traditional low dimensional data.*

1. Introduction

The Discriminative Patterns (DPs) can be used to find interesting information that occurs in a disproportionate frequency in one class in relation to the others. The DPs are techniques of data mining that evolved fast and under different names (Subgroups Discovery [Atzmueller 2015], Emerging Patterns [Dong and Li 1999] and Contrast Set [Bay and Pazzani 2001]), and afterwards the term DPs was formalized and generalized [García-Borroto et al. 2012][Liu et al. 2015].

This technique has been used to search knowledge in different areas, such as bioinformatics [Li e Wong 2001], marketing [Gamberger and Lavrac 2002], and e-learning [Romero et al. 2009] and its advantage over other techniques is the easiness in reading and interpretation of results after the mining [Vimieiro and Moscato 2014].

The approaches for the mining of DPs can be heuristic [Luna et al. 2013][Pontes et al. 2016], which does not guarantee an optimal response, that is, they do not list all relevant patterns. This result may in the disposal of significant data. In biomedical databases such discarding can be very important when genes that are considered of little significance are discarded, however they may be useful for different types of tissues. While exhaustive approaches guarantee an optimal response [Kavsek et al. 2003] [Atzmueller and Puppe 2006][Azevedo 2010] because they list all the relevant patterns of a solution. Most exhaustive approaches have been proposed with the objective of solving problems concerning traditional low dimension data. Thus, comparing these approaches in high dimensionality bases became important, because in previous works such comparison was not done and these approaches were not designed for this. In addition, the search for high dimensional bases is a recurring challenge mainly in biomedical domains [Vimieiro and Moscato 2014].

Therefore, this article aims at evaluating quantitatively the behavior of exhaustive mining algorithms of DPs in high dimensionality bases. For this purpose, experiments with the algorithms were carried out using real world, various domains,

high dimensionality and traditional low dimensionality databases. We evaluated the computational performance and the quality in patterns found through the quality measures, which permitted to verify the strong and weak points of each algorithm. With the results, it was also possible to classify the bases (*easy*, *difficult* and *intermediary*) used in the experiments in relation to the criteria used, in order to evaluate the characteristics that affected the performance of the algorithms tested.

The article is organized as follows: Section 2: basic concepts on DPs, Section 3: shows some related works, Section 4: describes the algorithms used, Section 5: presents the methodology used in the experiments, Section 6: the results and, at last, Section 7: presents the conclusion.

2. Discriminative Patterns

Consider a set of D data that contains transactions (samples) with 2 (*two*) classes, given as $D = D^+ \cup D^-$, assuming that D^+ are the transactions marked as positive, that is, the target of the search, and D^- are marked as negative, other transactions. A DP is formed by one or more items (characteristics). Each item consists of a pair (attribute, value). The universe of all items that occur in such transactions is given by $I = \{i_1, i_2, \dots, i_m\}$. For example, a DP of dimensionality three can be represented by $dp_3 = \{i_a, i_b, i_c\}$, where $dp_3 \subseteq I$.

Consider a hypothetical example of a treatment application of a disease. This application has information of each patient in a database: sex, symptoms, types of treatments adopted and condition. Considering that the objective is to identify characteristics of cured patient for the uncured ones. D would be the entire database, D^+ cured patient (*condition = cured*), D^- the other conditions.

Finally consider that a DP is given by a set of symptoms (S_1, S_2, S_3) and treatments (T_1, T_2). This DP has a growth rate of 10 from uncured to cured, that is, among all patients with the 3 symptoms and who received the 2 treatments the number of patients is 10 times higher than the uncured ones, indicating that the combination of treatments for this combination of symptoms may be more appropriate.

During the search process, the DPs are evaluated through quality measures to determine their relevance. There are several kinds of quality and it is important that the algorithm accepts the larger amount of measures to amplify the users' possibilities.

The measures used in this process are on Table 1, where TP , true positive (amount of transactions that satisfy the DPs conditions and belong to search target class) and FP , false positive (amount of transactions that satisfy the conditions but do not belong to the target class of the searches), $|D|$, $|D^+|$ and $|D^-|$ are the total amount of transactions, positive and negative.

The *support* measures the frequency of transactions correctly covered in relation to the total of transactions. The *confidence* (accuracy) shows the precision of transactions covered correctly in relation to the amount of transactions that satisfy the DPs' conditions. The *relative accuracy* (originality) describes the DPs whose distribution of search target class (D^+) is significantly different from set D . More details and other measures can be found in [Herrera et al. 2011][Liu et al. 2015].

The search for DPs can also be classified as local or global. In the local search all relevant restrictions specified by the user are imposed separately by each DP

(individually). In the global ones, they concentrate in finding a condensed set of significant and non-redundant DPs, and global restrictions can be specified for all the patterns set (collectively).

Table 1. Discriminative patterns quality measures

Quality measures	Description
$Sup = \frac{TP}{ D }$	Support [Herrera et al. 2011]
$Conf = \frac{TP}{TP + FP}$	Confidence [Herrera et al. 2011]
$WRAcc = \frac{TP + FP}{ D } \left(\frac{TP}{TP + FP} - \frac{ D^+ }{ D } \right)$	Relative DP accuracy [Herrera et al. 2011]

As to the numbers of DPs returned, the algorithms can return the k best DPs using the quality measures. Another way is through thresholds, where DPs are considered relevant according to one or more criteria, which satisfy a minimum value for the chosen quality measure, being the most common the minimum support and/or minimum confidence. Defining those thresholds is a difficult task, because if very low, thresholds do not represent a useful limitation, and if very high, they may not return any DP. Both forms can work together.

3. Related work

There are many exhaustive algorithms for the search for DPs [Kavsek et al. 2003][Fang et al. 2012][Liu et al. 2015]. Nevertheless, a problem found with the exhaustive approaches and of local search [Liu et al. 2015] was that most of the tests were held with bases of up to 445 attributes at the most, that is, not working with dimensionalities in the order of the milliards. Another issue is that the study [Fang et al. 2012] to work with high dimensionality with exhaustive algorithms in biomedical (microarray) bases (5981 attributes) remained at *support* thresholds (low values) and adaptation of *DiffSup* quality measure and experiments in an only real base. The third problem was the use of different quality measures for the thresholds of each algorithm, thus making it difficult the understanding and the operation by the user.

For this reason, we have chosen the exhaustive algorithms based on the following premises: (1) that they were of local search, since they are more used and are more suitable for exhaustive search [Liu et al. 2015], (2) that they had the same threshold (3) that they had the highest number of quality measures (*minimum support* and *minimum confidence*) as a threshold, since algorithms that offer more options are better adapted to the different situations [Pontes et al. 2016]. And finally (4) that it was not necessary to be done any alterations or adaptations in the algorithms, thus enabling a better equality comparison showing a wider range of results.

Finally, as to the quality of DPs, it was chosen the measure of quality of relative accuracy to quantify, for according to the latter [Kavsek et al. 2003][Herrera et al. 2011][Liu et al. 2015] it is the most adequate to describe the DPs. In the next section the chosen algorithms will be explained

4. Exhaustive Algorithms

4.1. APRIORI-SD

The APRIORI-SD [Kavsek et al. 2003] is an algorithm that adapted the APRIORI-C. Initially, the algorithm returns the DP candidates that satisfy the criteria of minimum support and minimum confidence. The APRIORI-SD calculates the relative accuracy for the candidates and orders them from better to worse. The best ones are selected. The remaining will be recalculated and the process is repeated until a stop criterion is met, for example, when there are no more candidates in the subset of DP candidates.

4.2. SD-Map

The SD-Map [Atzmueller and Puppe 2006] is an algorithm that adapted the algorithm FP-Growth and has as a characteristic working with bases with absent data and the possibility to use multiple criteria for the selection of DPs. Initially, minimum support DP candidates are selected. After that, the candidates are placed on a decreasing way in the tree based on the FP-Growth, and afterwards, it is computed the quality criteria (minimum confidence) for the candidates until the tree is finished, returning only the top-k best ones.

4.3. RCS

RCS [Azevedo 2010] is an algorithm that adapted the approaches of applications of the algorithm STUCCO and used the *depth first* search. To return candidates to the DPs, it also uses the minimum support and minimum confidence criteria through PEP (*Parent Equivalence Pruning*) technique, which cuts off all extensions of DPs candidates. At last, to determine its significance it is used the criteria of Fisher exact test, after they return all the best DPs found, since there is no top-k option.

5. Experiments

The experiments started with 30 database, where 20 low dimensionality databases extracted from the UCI - *Machine Learning Repository* [UCI 2016], described on Table 2, were used to compare the results with the 10 high dimensionality databases (microarrays), hereby called BIO, found in the package data microarray [Ramey 2016] for software R. All the bases are characterized by being real, traditional of literature and of several domains. For each database, a majority class was considered the target of the search (p) and the other examples labeled as negative (n), so we try to keep the same quantitative examples in both classes.

We have chosen only UCI bases with categorical attributes. However, the BIO bases had to be discretized. At first, the control attributes contained in the data had to be removed. Since each manufacturer uses different control attributes, a proper filter was applied for microarray bases in the *genefilter* package [Bioconductor 2016] of R. At last, the attributes were discretized with the first and forth quartiles, as usually happens with biomedical values under the first quartile were qualified as *low* ('b'), values above the third quartile, *high* ('a') and the remaining values as *average* ('m'). The final result is on Table 3, where A , $|I|$ and $Density$, are respectively the number of attributes, the number of possible attributes, the number of possible items for each attribute where $|I| \subseteq |D| \cap |A|$, and the density is $|\text{total number of items}| / (|D| * |A|)$.

The experiments were held in an 8 core Intel Xeon 3,4Ghz with 16GB RAM of dedicated memory. The experiments were divided into two phases. In the first phase, the 30 bases are submitted to the discriminative algorithms using as parameter the *minimum confidence* variation with the values (in percentage) 100, 80, 60, 40 and 20 and the *support* fixed (in percentage) in 10. In the second phase, the *minimum support* is varied with values 100, 80, 60, 40, 20 e 10 and *minimum confidence* fixed in 80. The fixed values of confidence and support were based on [Kavsek et al. 2003] and the standard values of the tools used. Thus, we performed a total of 990 tests. The tools, setting, execution and documentations are detailed at <https://github.com/aldenor23/comp>.

Table 2. UCI Databases description

Name	D	A	I	Density
audiology	226	69	154	44.81%
breast-cancer	286	9	41	21.95%
bridges-version2	105	12	191	6.28%
car	1728	6	21	28.57%
kr-vs-kp	3196	36	73	49.32%
lung-cancer	32	56	157	35.67%
molecular-biology-promoter	106	58	334	17.37%
monks-problems-1-train	124	6	17	35.29%
mushroom	8124	22	116	18.97%
nursery	12960	8	27	29.63%
postoperative-patient-data	90	8	23	34.78%
primary-tumor	339	17	37	45.95%
shuttle-landing-control	15	6	16	37.50%
solar-flare-2	1066	12	42	28.57%
soybean	683	35	99	35.35%
spect-test	187	22	44	50.00%
splice	3190	61	3465	1.76%
tic-tac-toe	958	9	27	33.33%
trains	10	32	77	41.56%
vote	435	16	32	50.00%

Table 3. BIO Databases description

Name	D	A	I	Density
alon	62	1000	3000	33.33%
burczynski	127	11107	33321	33.33%
chiaretti	128	6279	18837	33.33%
chin	118	11107	33321	33.33%
christensen	217	706	2118	33.33%
gravier	168	1452	4356	33.33%
nakayama	105	11099	33297	33.33%
sorlie	85	228	684	33.33%
tian	173	6279	18837	33.33%
yeoh	248	6279	18837	33.33%

The interruptions in the execution of the experiments occurred for two reasons: firstly, when arriving at limit timing (3 hours) as in [Vimieiro and Moscato 2014] or when there was a memory overload.

6. Results

We analyzed 14 aspects being that 7 concerning computational performance (time) and 7 in relation to the quality of the results (relevance) for each algorithm in each phase.

In phase 01, the algorithms were submitted to 30 bases and to the 5 parameters, which totaled 450 experiments (e) for each phase. Table 4 describes the performance for the first four aspects analyzed. We observed that for the BIO bases, all the experiments had interruptions in the execution, where, SD-Map and RCS had more interruptions due to memory overload and APRIORI-SD for time limit. Nevertheless, a particularity observed on the SD-Map (3 bases) and RCS (5 bases) was that both accused in the execution a very high number of transactions, being such bases characterized for being the ones with the higher number of attributes.

Concerning UCI, the first aspect describes the number of well-succeeded experiments (ec), that is, those that concluded the execution within the established time and did not overload the memory. The RCS algorithm had a better performance covering 80 experiments.

The second one is related to the general average time $\bar{t}(ec)$, which quantifies the average time (in seconds) in which each algorithm used in relation to the ec . The RCS also had a better performance, reaching a better timing in the processing.

The third aspect considers also the general average time. However, for a more equalitarian comparison, we have considered only the bases that had ec in all the algorithms $\bar{t}'(ec)$. Thus, the RCS had its performance amplified in relation to the previous aspect, being 30 times faster.

The fourth aspect analyzed the bases of isolated form $b(ec)$ for each algorithm, that is, in how many bases the algorithm had a better performance. The RCS and the SD-Map showed a larger quantity, being 8 bases for each.

Table 4. Algorithms performance for phase 01

	Algorithms	ec	$\bar{t}(ec)$	$\bar{t}'(ec)$	$b(ec)$
UCI	APRIORI-SD	70	366.741	417.487	2
	SD-Map	71	67.163	78.749	8
	RCS	80	66.421	2.585	8
BIO	APRIORI-SD	-	-	-	-
	SD-Map	-	-	-	-
	RCS	-	-	-	-

In relation to the fifth aspect analyzed, the total number of samples (D), it was noticed that the 10 bases with the largest number of samples (primary-tumor, 339, a nursery, 12960) the RCS algorithm, once more had a better performance by covering a larger amount of bases (90%) in relation to the APRIORI-SD and the SD-Map that has the same performance (80%).

In the sixth, the bases with higher density (spect-test, vote, kr-vs-kp e primary-tumor) the RCS also outraged converging in all (100%), and the remaining algorithms, only in the two last ones (50%). However, in relation to the number of items (I) the bases with the highest number (splice, molecular-biology-promoter e bridges-version2), the APRIORI-SD and the SD-Map converged in all (100%) while the RCS only in the last one (33.33%).

In general, the RCS is better in relation to the time in phase 01 (6 of 7 aspect) for traditional bases of low dimensionality, having difficulty of convergence in bases with quantitative of higher items (I) and number of attributes $|A|$ (audiology, splice,

molecular-biology-promoter and lung-cancer). However APRIORI-SD and SD-Map were able to converge in 2 of them (splice and molecular-biology-promoter).

Table 5 describes the performance in relation to the quality of the results (8th to 11th). It is considered now, among the successful experiments (*ec*), only the amount of experiments that returned DPs (*eq*), that is, the experiments with DPs equal to zero are excluded. In this eighth aspect, the RCS also had the quantitative superior to the others (60).

The ninth aspect considers the general average quality $\bar{q}(eq)$ (relative accuracy, between -0.25 and 0.25), that is, quantifies the average quality that each algorithm used in relation to the *eq*. The RCS was better for having more quality (0.077062).

The tenth also considers the general average quality. However, for a more equalitarian comparison, we considered only the bases that had *eq* in all the algorithms $\bar{q}'(eq)$. Thus, the RCS had a better performance in relation to the previous aspect and better than the other algorithms.

The following aspect analyzed the bases in an isolated form *b(eq)* for each algorithm, that is, in how many bases the algorithm had a better quality. The RCS and the SD-Map had a larger amount, being 8 bases for each.

Table 5. Algorithms quality performance for phase 01

	Algorithm	<i>eq</i>	$\bar{q}(eq)$	$\bar{q}'(eq)$	<i>b(eq)</i>
UCI	APRIORI-SD	56	0.073212	0.081084	8
	SD-Map	53	0.035123	0.030933	0
	RCS	60	0.077062	0.081267	8
BIO	APRIORI-SD	-	-	-	-
	SD-Map	-	-	-	-
	RCS	-	-	-	-

In the twelfth, in relation to the total number of samples (*D*), it was observed that with the 10 bases with the highest number of samples (primary-tumor, 339, a nursery, 12,960) the algorithm RCS showed the largest amount of experiments that returned DPs (90%) in relation to the SD-Map (72%) and the APRIORI-SD (64%).

In the thirteenth, the bases with higher density (spect-test, vote, kr-vs-kp e primary-tumor) in relation to the quality, the APRIORI-SD had a better performance, converging in three (75% of the experiments that returned DPs), and the remaining algorithms only in two (SD-Map 50% e APRIORI-SD 40%). At last, in the last aspect the relation of the incidence due to quality (*I*) in the bases with a larger amount (splice, molecular-biology-promoter and bridges-version2) the SD-Map was best converged in all (100% of the experiments that returned DPs) in relation to the APRIORI-SD (80%) and the RCS (33.33%).

Similar at time, the RCS is better in relation to the quality in phase 01 (6 of 7 aspect) for traditional bases of low dimensionality, having difficulty of convergence in bases with quantitative of higher items (*I*) and number of attributes $|A|$.

In phase 02, the algorithms were submitted to 6 parameters, which totalized 540 experiments (*e*). Table 6 describes the performance for the first four aspects analyzed. It was observed that number of well-succeeded experiments in the BIO (*ec*). For this first aspect, the RCS was better (104) in UCI bases. However, considering the BIO bases, the

APRIORI-SD was closer with 103. Considering the BIO bases, the APRIORI-SD had a better performance (35).

For the second aspect $\bar{t}(ec)$, the SD-Map had a better performance to make the experiments in a shorter period of time than the other UCI bases (14.630), being almost twice as fast as the second place. In relation to the BIO, the RCS had an apparent better performance. However, since it showed little convergence, a better comparison on this aspect was unfeasible.

The third analysis $\bar{t}'(ec)$, the RCS improved the performance in relation to the previous aspect for the UCI (1.800), for it came to be 7 times faster than the SD-Map. However, for BIO bases, the SD-Map is better (0.736).

The fourth aspect $b(ec)$, the APRIORI-SD had a better performance for being larger in quantity (10) followed by the SD-Map and RCS (5) for the UCI. However, for the BIO, the APRIORI-SD was better by having 6 isolated bases.

Table 6. Algorithms performance for phase 02

	Algorithms	ec	$\bar{t}(ec)$	$\bar{t}'(ec)$	$b(ec)$
UCI	APRIORI-SD	103	156.154	94.221	10
	SD-Map	98	14.630	13.898	5
	RCS	104	27.876	1.800	5
BIO	APRIORI-SD	35	1978.210	0.892	6
	SD-Map	29	80.450	0.736	4
	RCS	9	12.428	7.644	0

In the fifth aspect, it was observed the same that had occurred in the last phase for the UCI bases. The bases with the highest number of samples, the algorithm RCS had a better performance by covering this time the highest amount of bases (95%) in relation to the SD-Map (93.33%) and the APRIORI-SD (91.67%). However, for the BIO, there was no meaningful observation because they had very few samples.

In the sixth, the bases with highest density (spect-test, vote, kr-vs-kp e primary-tumor), the RCS also performed better converging in all (100%), and the other algorithms had the worst performance (APRIORI-SD 83.33% and SD-Map 70.83%). In the BIO bases all the algorithms were of the same density and analysis for this aspect was not possible.

In the number of items (I), the bases with a higher number (splice, molecular-biology-promoter and bridges-version2) the APRIORI-SD and the SD-Map had a better performance converging in all the (100%) while the RCS in 66.67%. In the BIO bases (burczynski, chin and nakayama) the APRIORI-SD had a better performance in 50% and the remaining algorithms did not show any responses.

We were also confirmed the observation of the latter phase in relation to the non-convergence of the bases for RCS with quantitative of higher items (I) and a larger number of attributes $|A|$ the APRIORI-SD and the SD-Map that managed to converge in 2 of them. The RCS is better in relation to the time in phase 02 (4 of 7 aspects). In the BIO bases, it was noticed that the APRIORI-SD had a better performance by managing to converge in all bases (10) at least once, and the SD-Map (7), managed to converge in bases with up to 6,280 attributes and the RCS (4) with up to 1,453 attributes.

In all experiments for this phase, there was a gradual increase in the execution time in relation to the support variation, occurring more frequently in the lowest measures (20 and 10) for the UCI and (40, 20 and 10) for the BIO.

Table 7 describes the performance in relation to the quality of the results (8th to 11th). In the eighth aspect (eq), the APRIORI-SD has a better performance by managing to have the highest number of experiments with return of DPs (35) for the UCI. However, the SD-Map is better (7) in BIO bases.

In the ninth aspect $\bar{q}(eq)$, the RCS has advantage for having a greater general average quality (0.077964) in the UCI bases. However, for the BIO and the APRIORI-SD bases have a better performance in relation to the others (0.095065).

In the following aspect $\bar{q}'(eq)$, again the APRIORI-SD has advantages as much to the UCI bases (0.089463) as to the BIO bases (0.092046). In the tenth $b(eq)$ it is where the APRIORI-SD has the greatest advantage by owning 11 bases with better quality for the UCI. For the BIO, it has a better performance, but with the same quantitative as of the SD-Map (3).

Table 7. Algorithms quality performance for phase 02

	Algorithms	eq	$\bar{q}(eq)$	$\bar{q}'(eq)$	$b(eq)$
UCI	APRIORI-SD	35	0.059352	0.089463	11
	SD-Map	28	0.044165	0.055500	0
	RCS	26	0.077964	0.083953	6
BIO	APRIORI-SD	3	0.095065	0.092046	3
	SD-Map	7	0.084583	0.086667	3
	RCS	0	-	-	0

In the twelfth, in relation to the total number of samples (D), it was observed that with the 10 bases with a higher number of samples for the UCI, the algorithm APRIORI-SD had a higher number of experiments that returned DPs (36.67%) against 33.33% for SD-Map and 31.67% for RCS. However, for the BIO, there was no meaningful observation for possessing very few sample variation.

In the thirteenth, the bases with a higher density for the UCI, the APRIORI-SD had a better performance, owning DPs in two bases (41.67% of the experiments that returned DPs) and the RCS and SD-Map with 29.17%. In the BIO bases all the algorithms were of the same density and analysis for this aspect was not possible.

Finally, in the last aspect the number of items (I) the bases with a higher number of UCI had the algorithms APRIORI-SD and SD-Map as the best ones. They converged at least once in all of the bases (33.34% of the experiments that returned DPs) and the RCS only in one (16.67%). As to the BIO bases, there was no meaningful quantitative for the evaluation of this aspect.

The APRIORI-SD is better in relation to the quality in phase 02 (6 of 7 aspect) for traditional bases of low dimensionality, having difficulty only $q(eq)$. APRIORI-SD had an apparently better quality results in BIO, however, there were very few quality results, because of the bases with high number of attributes $|A|$ that did not return DPs.

6.1. Data set classification

After all the experiments were carried out (total of 33 experiments per base) it was possible to classify the bases according to the degree of difficulty in the return of the

results in relation to the adopted criteria. With the classification it was possible to evaluate the characteristics that affect the performance of the algorithms, to add another element to the final comparison of the algorithms, and also to help in future uses of the bases in DPs. Table 8 shows the classification of the bases. The \overline{ec} is the average of successful experiments (*ec*) for the algorithms. The algorithms in the *easy* bases had a behavior where $\overline{ec} = 11$, in the bases considered as *difficult*, the $\overline{ec} \leq 5$ and in the bases considered *intermediate*, the algorithms stayed between 11 and 5.

As to the quality criterion, the \overline{eq} is the average of the experiments that returned DPs for each algorithm. The *easy* bases characterized as having $\overline{eq} \geq 5$, in the *difficult* bases the $\overline{eq} \leq 1$ and in bases considered *intermediate*, the algorithm remained between 5 and 1.

Table 8. Classification data sets according experiments

	Base	\overline{ec}	\overline{eq}
<i>Easy</i>	breast-cancer	11	6,67
	bridges-version2	11	7
	car	11	6,67
	mushroom	11	7,67
	nursery	11	6,67
	primary-tumor	11	8
	solar-flare-2	11	5
	tic-tac-toe	11	5,67
	vote	11	7,33
	<i>Difficult</i>	spect-test	6
trains		5	0
sorlie		4	1
christensen		3,67	0,33
alon		3,33	0,67
gravier		3,33	0,67
chiaretti		2,33	0
tian		2,33	0,33
yeoh		2,33	0,33
lung-cancer		1,33	0
audiology		1	0
burczynski		1	0
chin		1	0
nakayama		1	0
<i>Intermediate</i>		monks-problems-1-train	11
	postoperative-patient-data	11	3
	shuttle-landing-control	11	2
	molecular-biology-promoters	8,33	4,33
	splice	8,33	4,33
	soybean	6,33	4
	kr-vs-kp	6	2,67

There were exceptions to the classification due to our having considered the most important quality criterion. Firstly, the spect-test base is on the threshold to be classified as *intermediate*. However, its quality was *difficult* so it was decided to have this classification. Finally, the bases monks-problems-1-train, postoperative-patient-data and shuttle-landing-control, could be classified as *easy*. However, their quality is *intermediate*.

It is noticed that all BIO bases were classified as difficult to return, confirming that high dimensionality affects the performance of the algorithms. Another characteristic is that the bases classified as easy are of low dimensionality ($A < 22$).

7. Conclusions

In the UCI bases, it is evaluated that when the *confidence* quality measure is varied and bases of few attributes and low numbers of items are used, the RCS has a better performance (12 of 14 criteria), in relation of quantity of converged experiments (*ec*), and time $\bar{t}(ec), \bar{t}'(ec), b(ec)$, number of samples $|D|$ and density. In relation of quality of the results *eq*, $\bar{q}(eq), \bar{q}'(eq), b(eq)$, number of samples $|D|$ and density. Finally, such results are achieved in bases classified as *easy* and *intermediate*.

We also concluded when the *support* measure is varied, independently of the quantitative of attributes, the APRIORI-SD has a better performance (8 of 14 criteria). In relation to time $b(ec)$ and higher number of items $|I|$. As to the quality, it had superior performance in *eq*, $\bar{q}'(eq), b(eq)$, number of samples $|D|$, density and higher number of items $|I|$. Finally, it reaches such results in all of the base classifications. Thus, we have noticed in this phase that besides having an inferior time performance as compared to the others, the APRIORI-SD returned more DPs and with higher qualities.

In the BIO (high dimensionality) bases, in the *confidence* measure, it was not possible to define which one was the best algorithm because no satisfactory result was found in these kinds of base. As to the *support* measure, there were few results with quality, despite an advantage of the APRIORI-SD in 6 criteria. It was observed that all BIO bases were classified as *difficult* due to the performance of the algorithms, that is, most of the experiments the algorithms could not finish within the time and when they finish, few returned DPs. So we come to conclude that such algorithms, in a pure manner, are not the most appropriate of the high dimensionality data. New studies should be done to improve the performance of these algorithms.

As future works, we intend to adapt the union of the thresholds with the top-k in the RCS algorithm, to improve its appearance with the *support* in bases of low dimensionality. Another proposal is to improve the suitability of APRIORI-SD for high dimensionality bases, since among the algorithms it had some results with DPs. The third possibility is to adapt/modify other exhaustive algorithms to the same criteria adopted in this work to extend the comparison.

References

- Atzmueller, M. (2015) “Subgroup Discovery”, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no.1, pp35-49.
- Atzmueller, M. and Puppe, F. E. (2006) “SD-Map - a fast algorithm for exhaustive subgroup discovery”, *Proceedings of the 17th European conference on machine learning and 10th European conference on principles and practice of knowledge discovery in Databases*, Vol. 4213, pp. 6–17.
- Azevedo, P. (2010) “Rules for Contrast Sets”, *In Intelligent Data Analysis*, 14, pp. 623-640.
- Bay, S. D. and Pazzani, M. J. (2001) “Detecting group differences: Mining contrast sets”, *Data Mining and Knowledge Discovery*, vol. 5, no.3, pp. 213-246.

- Bioconductor (2016), Software Package *genefilter* v1.56.0 [Online] Disponível em: <http://bioconductor.org/packages/release/bioc/html/genefilter.html>
- Dong, G. and Li, J. (1999) “Efficient mining of emerging patterns: Discovering trends and differences,” in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 43-52.
- Fang, G., Pandey, G., Wang, W., Gupta, M., Steinbach, M. and Kumar, V. (2012) “Mining low-support discriminative patterns from dense and high-dimensional data”. *IEEE Transactions on Knowledge and Data Engineering*. v. 24, n. 2, p. 279-294.
- Gamberger, D. and Lavrac, N. (2002) “Expert-guided Subgroup Discovery: methodology and application” *J.Artif. Int. Res*, v.17, n. 1, p.501-527.
- García-Borroto, M., Martínez-Trinidad, J. and Carrasco-Ochoa, J. (2012) “A survey of emerging patterns for supervised classification”. *Artificial Intelligence Review*, v. 42, n. 4, p. 705 – 721.
- Herrera, F., Carmona, C. J., González, P. and Del Jesus, M. J. (2011) “An overview on subgroup discovery: foundations and applications,” *Knowledge and information systems*, vol. 29, no. 3, pp. 495–525.
- Kavsek, B., Lavrac, N. and Jovanoski, V. (2003) “APRIORI-SD: Adapting Association Rule Learning to Subgroup Discovery”. In: *Proc. 5th Intl. Symposium on Intelligent Data Analysis*, Springer Verlag, 230–24.
- Li, J. and Wong, L. (2001) “Emerging patterns and gene expression data”. *Genome Informatics*, v. 12, p. 3-13.
- Liu, X., Wu, J., Gu, F., Wang, J. and He, Z. (2015) “Discriminative pattern mining and its applications in bioinformatics,” *Briefings in bioinformatics*, v.16, n.5, p. 884-900.
- Luna, J. M. J., Romero, R., Romero, C. and Ventura, S. (2013) *Discovering subgroups by means of genetic programming*. Springer.
- Pontes, T., Vimieiro, R. and Ludermir, T. B. (2016) “SSDP: A Simple Evolutionary Approach for Top-K Discriminative Patterns in High Dimensional Databases.” *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*. IEEE.
- Ramey, J. (2016) The *datamicroarray* package. [Online]. Disponível em: <https://github.com/ramhiser/datamicroarray>
- Romero, C., Gonzalez, P., Ventura, S., Del Jesus, M. J. and Herrera, F. (2009) “Evolutionary algorithms for subgroup discovery in e-learning: a practical application using moodle data”. *Expert Systems with Applications*, v.36, n.2, p.1632-1644.
- UCI – Machine Learning Repository (2016), Center for machine Learning and Intelligent Systems, University of California, Irvine [Online]. Disponível em: <https://archive.ics.uci.edu/ml/datasets.html>.
- Vimieiro, R. and Moscato, P. (2014) “A new method for mining disjunctive emerging patterns in high-dimensional datasets using hypergraphs”. *Information Systems*, v. 40, p. 1 – 10.

Protótipo de um Método de Classificação por Descrição Mínima

Breno Lima de Freitas, Tiago A. Almeida, Renato M. Silva

¹Departamento de Computação (DComp)
Universidade Federal de São Carlos (UFSCar) – Sorocaba – SP – Brasil

brenolimadefreitas@gmail.com, talmeida@ufscar.br,
renatoms@dt.fee.unicamp.br

Abstract. *For the last decades, many Machine Learning methods have been proposed aiming categorizing data. Given many tentative models, those methods try to find the one that fits the dataset the best by building a hypothesis that predicts unseen samples reasonably well. One of the main concerns in that regard is selecting a model that performs well in future samples not overfitting on the known data. In this paper, we introduce the initial prototype of a classification method based on the minimum description length principle, which naturally offers a tradeoff between model complexity and data fit. The proposed method is multiclass, online and is generic in the regard of data representation. Although the proposed method being an initial prototype, the conducted experiments on well-known public datasets showed that it performs as good as traditional classification approaches.*

Resumo. *Ao longo das últimas décadas, diversos métodos de aprendizado de máquina vêm sendo propostos com o intuito de classificar dados. Entre os modelos candidatos, procura-se selecionar um que se ajuste bem aos dados de treinamento, criando uma hipótese que faça boas previsões em amostras não analisadas anteriormente. Um dos maiores desafios é selecionar um modelo, cuja hipótese não seja sobre-ajustada aos dados conhecidos, sendo genérica o suficiente para boas previsões futuras. Neste trabalho, é apresentada uma proposta inicial de um método de classificação baseado no princípio da descrição mais simples, que efetua uma troca benéfica entre a complexidade do modelo e o ajuste aos dados. O método proposto é multiclasse, incremental e pode ser usado em dados com atributos categóricos e numéricos. Apesar de ser um protótipo inicial, experimentos com bases de dados públicas e bem conhecidas mostraram que ele é competitivo com métodos consolidados de classificação.*

1. Introdução

Ao longo das últimas décadas, a quantidade de dados tem aumentado em ritmo sem precedentes, o armazenamento e manipulação das informações tem se tornado mais acessível e o poder de processamento computacional tem aumentado. Ao mesmo tempo, tem crescido a necessidade de extrair informações dos dados de forma automática. Nesse contexto, diversos métodos de aprendizado de máquina vêm sendo propostos com a finalidade de categorizar (ou classificar) dados. Eles baseiam-se na seleção de um modelo, dada uma hipótese, que tem como objetivo ajustar-se a um conjunto de dados conhecidos para que possa computar uma saída (classe) para uma amostra nunca observada. É desejável que o modelo escolhido tenha uma alta capacidade de *generalização*, isto é, seja capaz de produzir uma saída considerada adequada e consistente à solução do problema, tanto para as amostras que compõem o conjunto de dados de treinamento, quanto para amostras ainda não observadas.

Dado o grande número de possíveis modelos que podem ser usados para fazer predições, os métodos de classificação utilizam critérios de seleção distintos (*e.g.*, probabilidade, otimização, distância) para tentar selecionar o melhor modelo possível. Uma possível estratégia de seleção de modelos é a utilização de critérios baseados na *navalha de Occam*. Esse é um princípio filosófico, cunhado por William Occam, um filósofo e frade inglês, baseado na seguinte ideia: “entidades não devem ser multiplicadas além do necessário” – uma crítica à filosofia escolástica que tratava a realidade com teorias muito complexas. No contexto de aprendizado de máquina, a navalha de Occam é utilizada para que, na escolha de um modelo, dada a presença de múltiplas opções que descrevem o mesmo problema, seja dada preferência ao modelo mais simples.

Uma das mais conhecidas formalizações para o uso da navalha de Occam no problema de seleção de modelos foi definida por Rissanen [Rissanen 1978] por meio do princípio da descrição mais simples (*minimum description length* – MDL). Tal princípio também está enraizado nas ideias da complexidade de Komolgorov [Kolmogorov 2016] e define que o modelo que melhor se adapta aos dados e possui menor tamanho de descrição – portanto, o menos complexo – deve ser o modelo selecionado. Deste modo, o MDL elege modelos que preservam um equilíbrio favorável entre a sua capacidade de ajuste aos dados de treinamento e sua complexidade, evitando naturalmente o problema de sobreajustamento (*overfitting*).

O MDL foi utilizado em diferentes contextos de aprendizado de máquina, tais como: geração de árvores de decisão [Ross Quinlan e Rivest 1989, Kononenko 1998], redes Bayesianas [Lam e Bacchus 1994, Friedman et al. 1997] e categorização de textos [Bratko et al. 2006, Braga e Ladeira 2008, Silva et al. 2017]. O método MDLText, introduzido por [Silva et al. 2017], é um bom exemplo de sucesso da utilização de métodos baseados no princípio MDL em problemas de classificação. Porém, ele é especializado em problemas de categorização de texto.

Neste trabalho, é oferecida uma proposta inicial de um método de classificação que incorpora as vantagens teóricas oferecidas pelo princípio MDL. O método proposto, nomeado *Aprendiz de Descritores de Mistura Gaussiana (Gaussian Mixture Descriptor Learner – GMDL)*, é genérico e pode ser utilizado em qualquer problema de classificação binária ou multi-classe, cujos dados possam ser representados por atributos categóricos ou numéricos.

O GMDL, por ser baseado no princípio MDL, é robusto a um dos principais problemas que afeta vários métodos de classificação da literatura: o sobreajustamento aos dados. Além disso, ele cria seu modelo de predição de maneira incremental. Portanto, diferente da maioria dos métodos de classificação tradicionais, ele não é limitado ao cenário de aprendizado em *batch*, onde todos os dados de treinamento devem ser apresentados de uma única vez. O GMDL também pode ser utilizado em problemas que possuem um fluxo de dados contínuo, onde o modelo de predição precisa ser atualizado quando novas amostras são apresentadas para treinamento. Atualmente, na era do *big data*, onde muitos problemas contêm uma quantidade massiva de dados, métodos que geram modelos de predição de forma incremental são bastante desejáveis, pois essa característica os torna naturalmente escaláveis [Hoi et al. 2014].

O restante desse trabalho está organizado da seguinte forma: a Seção 2 apresenta os principais conceitos básicos sobre o princípio MDL. O método de classificação proposto neste trabalho é apresentado na Seção 3. Na Seção 4, é descrita a metodologia experimental. A Seção 5 apresenta os resultados obtidos. Por fim, a Seção 6 oferece as conclusões e direcionamentos para trabalhos futuros.

2. O princípio da descrição mais simples

Rissanen [Rissanen 1978, Rissanen 1983] formalizou o princípio da descrição mais simples (do inglês, *Minimum Description Length* – MDL), o qual dita que no problema de seleção de modelos, aquele que possui o menor tamanho de descrição deve ser priorizado. Este princípio, oriundo da Teoria da Informação, delinea que quanto mais se conhece os dados, maior será a regularidade descoberta e, portanto, mais pode-se comprimí-los [Grünwald 2005].

O MDL possui raízes na complexidade de Kolmogorov [Kolmogorov 2016], definida como o menor tamanho de um programa que imprime uma dada sequência e finaliza. Portanto, quanto menor a complexidade de Kolmogorov de um modelo, mais conhecimento esse modelo possui sobre a sequência sendo codificada, já que existe uma codificação menor dos dados. Logo, o modelo com menor complexidade de Kolmogorov deve ser selecionado para representar a sequência [Barron et al. 1998]. A unificação dos conceitos da Navalha de Occam e da complexidade de Kolmogorov oferecem ao MDL um equilíbrio benéfico entre a seleção da complexidade do modelo e seu ajuste aos dados, o que evita, portanto, um modelo complexo sobreajustado aos dados [Grünwald 2000].

O MDL procura por uma codificação C de um conjunto de dados D que consiga descrevê-lo de maneira única da menor forma possível. Isto é, procura-se um C no qual $L_C(D)$ seja o menor possível, onde L_C é uma função que descreve o tamanho dos dados codificados com o auxílio de C . No contexto desta pesquisa, entre todas as possíveis codificações, foram utilizadas aquelas descritas na base binária, usando *bit* como medida de informação. Como tratam-se de codificações binárias, no restante do texto será utilizado *log* para representar o logaritmo na base 2.

O MDL foi originalmente cunhado como um método de duas partes – também chamado de MDL primitivo (do inglês, *crude MDL*) - onde, dado um conjunto de modelos candidatos M , o modelo M que deve ser escolhido é aquele que minimiza a seguinte equação:

$$M_{MDL} := \arg \min_{M \in \mathcal{M}} [L(M) + L(D|M)]. \quad (1)$$

É possível mostrar que o tamanho ótimo de descrição é limitado inferiormente por $-\log P(x^n)$ para uma dada sequência de símbolos $x^n \in \mathcal{X}^n$ em um alfabeto \mathcal{X} . Uma prova para esta afirmação pode ser encontrada em [MacKay 2003, Seção 5.3]. Deste modo, obtém-se uma boa estimativa para $L(D|M)$, uma vez que existe uma codificação C , com L_C associado, tal que para todo $x^n \in \mathcal{X}^n$, $L(x^n) = \lceil -\log P(x^n | M) \rceil$, onde $P(\cdot | M)$ é a probabilidade associada a uma função de código para \mathcal{X}^n . Este método é aceitável para a segunda parte da Equação 1, uma vez que (i) a log-semelhança é um método estatístico padrão para medir o quão bem um modelo está ajustado à hipótese e (ii) se P gera \mathcal{X}^n , então L_C é ótima [Grünwald 2005]. No entanto, ainda não é claro como codificar o próprio M .

[Rissanen 1983, Barron e Cover 1991, Barron et al. 1998, Hansen e Yu 2001] deram os primeiros passos para refinar o MDL usando o que é chamado de codificação universal, abandonando o MDL de duas partes em favor de uma descrição mais formal e direta [Grünwald 2005]. Um fator $\bar{L}_{\mathcal{L}}$ é uma codificação universal se comprime todo símbolo de uma sequência quase tão bem quanto a codificação em \mathcal{L} que mais comprimiria tal sequência. Idealmente, busca-se uma codificação que seja universal no sentido que possa comprimir ao máximo todas as sequências dado o conjunto das funções de tamanho de código. No entanto, como mostrado por [Grünwald 2005], não é possível obter tal função. Portanto, busca-se uma codificação que seja aproximadamente tão boa quanto a melhor, que comprime todas as sequências possíveis. Em [Grünwald 2005, Proposição

2.14] foi mostrado que é possível encontrar uma distribuição, que aproxime razoavelmente bem os dados de uma codificação, utilizando uma distribuição normalizada baseada na complexidade do modelo utilizado para a codificação. Deste modelo surge o *MDL refinado*, que pode ser matematicamente formalizado pela seguinte equação:

$$M_{MDL} := \arg \min_{M \in \mathcal{M}} \bar{L}(D|M). \quad (2)$$

Não existe um consenso de como usar de maneira prática a codificação universal empregada pelo *MDL refinado*. Sendo assim, no método proposto neste trabalho, o tamanho de descrição calculado pela Equação 2 considera a distribuição de probabilidade dos dados, conforme detalhado na seção apresentada a seguir.

3. Aprendiz de Descritores de Mistura Gaussiana

Um dado (ou amostra) \vec{x} pode ser representado por um conjunto de atributos reais (x_1, \dots, x_n) . Utilizando os conceitos do MDL, procura-se pelo modelo M que mais comprima \vec{x} . Cada modelo M induz uma distribuição de probabilidade P para uma dada codificação de dados. É conhecido que a menor codificação possível para uma determinada codificação induzida por uma distribuição de probabilidade P é definida por $[-\log P]$ [MacKay 2003, Seção 5.3].

Idealmente, se a distribuição de probabilidade real que descreve um conjunto de dados for conhecida previamente, sob uma perspectiva Bayesiana, é possível prever com perfeição uma amostra não rotulada. No entanto, em aplicações reais isso não é possível, sendo necessário fazer uma estimativa de tal distribuição. Portanto, deseja-se alguma estimativa de distribuição de probabilidade que evite discretizações e seja incremental, para que seja mantida a habilidade do método de aprender com novas amostras para um fluxo contínuo de dados. Deste modo, neste trabalho, a Equação 2 foi utilizada no protótipo de um método de classificação baseado em distribuições de probabilidade, nomeado Aprendiz de Descritores de Mistura Gaussiana (*Gaussian Mixture Descriptor Learner* – GMDL), onde cada modelo induz uma distribuição para cada uma das classes do problema sendo avaliado.

No GMDL, para estimar uma função de densidade de probabilidade para o conjunto de dados, foi utilizada a forma clássica e paramétrica que é assumir que a distribuição é uma mistura Gaussiana [McLachlan e Peel 2004]. No entanto, tal técnica, no geral, depende da definição prévia dos parâmetros da mistura [McLachlan e Peel 2004, Zivkovic e van der Heijden 2004]. Estimar tais parâmetros não é uma tarefa trivial. O uso, por exemplo, do número errado de componentes, pode fazer com que o método não represente bem a função real [Kristan et al. 2011].

[Silverman 1986] cunhou um método não-parametrizado para estimar a função de densidade de probabilidade de variáveis aleatórias chamado de *estimativa de densidade de kernel* (EDK). Este método possui uma função chamada de *kernel*, que integra a um e tem média zero. Ele também utiliza um parâmetro b chamado de *largura de banda*, cujo papel é suavizar a soma dos *kernels* do estimador da distribuição de probabilidade. A seleção do *kernel* e da largura de banda tem grande influência na estimativa da função [Silverman 1986]. Uma escolha clássica de *kernel* é o Gaussiano, que mapeia a entrada para uma saída normalmente distribuída.

O EDK possui uma das características desejadas para a distribuição apresentada: não é necessária a discretização dos dados a priori. Com isso, é possível estimar as distribuições de probabilidade sem sofrer as perdas de dados oriundas de discretizações e manter uma estimativa da distribuição real dos dados. Essa é uma característica desejada

para o método que está sendo proposto nesse trabalho, uma vez que deseja-se que ele seja genérico e possa ser usado em problemas com atributos numéricos sem a necessidade de realizar discretizações. Porém, também deseja-se que o GMDL gere seu modelo de predição de forma incremental. Portanto, para que o EDK seja incorporado no método que está sendo proposto, é necessário que o EDK também seja incremental.

3.1. Estimativa de Densidade de Kernel Online

Uma das grandes dificuldades em se transformar o EDK em um método incremental é que o número de componentes que o método utiliza para a estimativa da distribuição cresce linearmente para cada nova amostra apresentada [Kristan et al. 2011]. Isso deve-se ao fato dele precisar manter informações o suficiente para generalizar para novas amostras sem a necessidade de coletar novamente informações das amostras vistas anteriormente [Ferreira et al. 2016]. Recentemente, [Kristan et al. 2011] propuseram um método de estimativa de densidade de *kernel* totalmente incremental, chamado de oKDE. Esta abordagem é ideal para contemplar a segunda característica desejada para uma estimativa de função de distribuição de probabilidade.

A ideia principal do oKDE é que, ao invés de tentar construir diretamente a distribuição alvo, mantém-se uma *distribuição amostral* não-parametrizada. Tais distribuições são construídas por um conjunto de amostras tratadas como funções Delta-Dirac, que podem ser definidas pela equação apresentada a seguir:

$$p_{DD}(x) := \lim_{\sigma \rightarrow 0} \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (3)$$

Cada distribuição amostral é, portanto, um conjunto de funções de densidade infinita em um único ponto, sendo que uma distribuição amostral pode ser vista como uma mistura Gaussiana. Neste caso, a largura de banda é dada pela matriz de covariância da distribuição.

Uma vez que manter uma função Delta-Dirac para cada observação não é adequado para um cenário de aprendizado incremental, o número de componentes no oKDE é reduzido por meio de um algoritmo de agrupamento que aproxima uma distribuição Gaussiana dados outros pontos. Esse agrupamento se dá pela redução de um conjunto n -dimensional para um m -dimensional, tal que $m < n$, onde procura-se por uma estimativa de distribuição que não ultrapasse uma dada taxa de erro D_{th} . Durante a estimativa do oKDE, também é utilizado um parâmetro f chamado de *fator de esquecimento*. Ele é utilizado para atribuir um peso às amostras antigas, diminuindo a influência delas em um fluxo de dados – ideal para cenários onde o fluxo de dados é temporal. Um fator de esquecimento igual a um significa que os pesos das amostras não se diferenciam por sua posição temporal.

O segundo ponto importante na descrição do oKDE é a estimativa da largura de banda ótima. A qualidade da estimativa de banda tem impacto direto na qualidade da distribuição aproximada. A estimativa da largura de banda pode ser vista como um problema de otimização onde deseja-se minimizar a distância da distribuição aproximada para a distribuição real [Lüthke 2013]. Usualmente, nesse cenário, utiliza-se a divergência de Kullback-Leibler como medida de distância [Kristan et al. 2011]. No entanto, a distribuição original não é conhecida. Para tais cenários, uma abordagem tradicional é utilizar o erro quadrático médio assintótico integrativo (do inglês, *asymptotic mean integrated squared error – AMISE*) [Wand e Jones 1994]. Os autores então a utilizam para definir a melhor largura de banda [Lüthke 2013].

[Ferreira et al. 2016] cunharam uma versão do oKDE, chamada de *xokde++*, que procura sanar problemas de inconsistência numérica e propor uma abordagem computacional mais robusta em termos de uso de memória e processamento. No *xokde++* também é mantida uma distribuição normalizada, isto é, todas as estimativas tem densidade limitada superiormente por um – uma propriedade extremamente útil, uma vez que pode ser mapeada para uma estimativa de probabilidade. O método que está sendo proposto neste trabalho utiliza essa versão melhorada do oKDE. No restante do texto, por simplicidade, o *xokde++* será referenciado como oKDE.

3.2. Predição incremental de amostras

O oKDE provê uma maneira não-parametrizada e incremental de estimar uma função de densidade para um conjunto de dados. Utilizando o oKDE, pode-se estimar densidades para compor os modelos avaliados pelo GMDL. Com esta aproximação espera-se que, com o número crescente de amostras, obtenha-se uma estimativa cada vez mais fidedigna da distribuição real.

É possível definir a função de tamanho de descrição \hat{L} como a soma das descrições de seus atributos quando codificados pela aproximação de suas funções de densidade p' para cada classe. Como mostrado em [MacKay 2003, Seção 5.3], sabe-se que a menor codificação que pode ser obtida é delimitada superiormente pelo logaritmo de sua função de probabilidade. Portanto, \hat{L} pode ser formalmente definido como:

$$\hat{L}(\vec{x}|c) := \sum_{i=1}^n [-\log p'_{(i,c)}(x_i)]. \quad (4)$$

Na Equação 4, \vec{x} é um vetor (x_1, \dots, x_n) de atributos e $c \in K$, a classe avaliada. Note que p' pode tender ao infinito, quando avalia-se uma função Delta-Dirac, e pode ser zero, caso não haja amostras o suficiente para medir a densidade em um ponto. Portanto, a Equação 4 pode ser reescrita como:

$$\hat{L}(\vec{x}|c) := \sum_{i=1}^n [-\log p_{(i,c)}(x_i)], \quad (5)$$

onde o valor de $p_{(i,c)}$ pode ser calculado da seguinte forma:

$$p_{(i,c)} := \begin{cases} 2^{-\Omega} & p'_{(i,c)} \rightarrow \infty \vee p'_{(i,c)} = 0 \\ p'_{(i,c)} & c.c. \end{cases} \quad (6)$$

Na Equação 6, Ω é um meta-parâmetro que funciona como regularizador nos casos onde não há informação o suficiente para computar-se uma probabilidade. É possível também alterar o fator de esquecimento f original no oKDE como um meta-parâmetro de p' . Para simplificar a notação, f será omitido.

O fator $L \in [0, 1]$ será definido como a versão normalizada de \hat{L} , isto é, o vetor resultante da divisão de cada elemento de \hat{L} pela soma de todos os elementos. Tal versão provê melhor estabilidade numérica e, convenientemente, uma pseudo-probabilidade quando avaliada como $1 - L(\vec{x})$.

Deste modo, a primeira versão do GMDL pode ser definida pela seguinte equação:

$$GMDL(\vec{x}) := \arg \min_{c \in K} L(\vec{x}|c). \quad (7)$$

Diante do exposto, o GMDL pode ser resumido da seguinte forma: dado um conjunto de possíveis classes $c_1, c_2, \dots, c_{|K|}$, quando um dado \vec{x} desconhecido for apresentado, ele será rotulado com a classe c_k que possui o menor tamanho de descrição $L(\vec{x}|c_k)$ em relação a \vec{x} .

3.3. Suavização de funções Delta-Dirac e distribuições degeneradas

Originalmente, o método proposto por [Kristan et al. 2011] (oKDE), que foi incorporado ao GMDL, era suscetível às distribuições degeneradas, isto é, distribuições com baixa variância.

O método refinado por [Ferreira et al. 2016] tentou mitigar esse problema e melhorar a estabilidade numérica. Os autores utilizaram o logaritmo da matriz de covariância para suas computações. Logaritmos são amplamente utilizados em Ciência da Computação para evitar problemas multiplicativos, uma vez que $\log(ab) = \log a + \log b$, o que mantém a estabilidade numérica na operação. Porém, $\log 0$ tende a infinito, uma indefinição teórica que faz com que o método seja propenso a erros. A técnica para minimizar o problema de distribuições degeneradas, proposta por [Ferreira et al. 2016], é computar sua decomposição em autovalores e autovetores, além de analisar seus autovalores procurando por aqueles menores que 10^{-9} , os quais são corrigidos por 1% da média dos autovalores. Contudo, é importante observar que esta técnica não soluciona o problema quando há apenas uma dimensão envolvida.

Para solucionar tal problema, um ruído $\mathcal{N}(0, \bar{\sigma}^2)$ pode ser adicionado. É fato conhecido que a soma de duas variáveis aleatórias normalmente distribuídas também é normal. Esta perturbação mantém o centro da distribuição e altera apenas seu desvio padrão. A primeira característica é muito útil, uma vez que, para funções Delta-Dirac, o ponto central de tal função continua possuindo o maior acúmulo de densidade. No entanto, alterar o desvio padrão também altera a densidade dos 68% dos dados em volta da média. Deste modo, o ruído é usado seletivamente quando a variância de um atributo em uma dada classe tornar-se menor do que o limiar de 10^{-9} definido por [Ferreira et al. 2016]. Este processo, dado o meta-parâmetro $\bar{\sigma}^2$, que determina o desvio padrão do ruído a ser aplicado, pode ser definido por:

$$p'_{(i,c)} \leftarrow \begin{cases} x_i & \sigma_{ic}^2 \geq 10^{-9} \\ x_i + \sum_{n \sim \mathcal{N}(0, \bar{\sigma}^2)} n, \sigma_{ic}^2 < 10^{-9} & c.c. \end{cases}$$

A computação incremental da variância é feita por uma aproximação baseada no acúmulo da média. Este método foi criado por Welford [Welford 1962] e posteriormente refinado por [Ling 1974] e [Tony F. Chan, Gene H. Golub 1983]. Ele se baseia em um fluxo contínuo de dados de uma variável aleatória. No caso específico do GMDL, esse fluxo é dado pelas amostras para cada atributo i da classe c .

4. Análise experimental

Para avaliar o desempenho do protótipo proposto do GMDL, foram realizados experimentos com bases de dados tradicionais, mostradas na Tabela 1, que estão disponíveis publicamente no site da UCI¹ [Lichman 2013]. Essas bases foram escolhidas por serem popularmente conhecidas, consolidadas e amplamente utilizadas em diversos trabalhos de aprendizado de máquina.

¹UC Irvine (UCI) Machine Learning Repository. Disponível em: <http://archive.ics.uci.edu/ml/index.php>. Acessado em 28/06/2017.

Tabela 1. Informações das bases de dados utilizadas nos experimentos.

Base de dados	Tamanho		Composição das classes		Composição dos atributos	
	m	n	$ K $	Amostras por classe	Dados contínuos	Dados discretos
BreastCancer	569	30	2	212, 357	✓	
CoverType	581012	10	7	2747, 9493, 17367, 20510, 211840, 283301, 35754		✓
Iris	150	4	3	50, 50, 50	✓	
Letter	20000	16	26	734, 734, 736, 739, 747, 748, 752, 753, 755, 758, 761, 764, 766, 768, 773, 775, 783, 783, 786, 787, 789, 792, 796, 803, 805, 813		✓
Skin	245057	3	2	50859, 194198	✓	
Wine	178	13	3	48, 59, 71	✓	✓
WineRed	1599	11	6	10, 18, 53, 199, 638, 681	✓	
WineWhite	4898	11	7	5, 20, 163, 175, 880, 1457, 2198		✓

Na Tabela 1, m corresponde ao número total de amostras, n é o número de atributos e $|K|$ é a quantidade de classes. É importante destacar que as bases escolhidas possuem características distintas, o que aumenta o escopo de validação do método proposto. Algumas delas tem um baixo número de amostras, enquanto outras superam 200 mil amostras. Algumas são binárias, enquanto outras são multiclasse, sendo que a maior delas tem 26 classes. Além disso, foram avaliadas bases de dados balanceadas e outras com alto grau de desbalanceamento. Por fim, foram avaliadas bases de dados com atributos contínuos e discretos.

4.1. Pré-processamento e avaliação

Em todos os experimentos, as amostras foram normalizadas para possuírem média zero e desvio padrão unitário. Além disso, para avaliar o desempenho de cada método, dividiu-se, de forma estratificada, cada base de dados em 80% para treinamento e validação e 20% para teste. As medidas de desempenho utilizadas foram a micro e a macro F-medida.

Apesar do método proposto neste trabalho ser naturalmente incremental, o cenário escolhido para os experimentos foi o aprendizado em *batch*, em que todos os dados de treinamento são apresentados de uma única vez aos métodos de classificação. O objetivo é analisar o método proposto no cenário clássico de aprendizado e compará-lo com técnicas tradicionais e consolidadas, sendo que a maioria deles não suporta aprendizado incremental.

4.2. Métodos

Para avaliar o desempenho do método proposto, seus resultados foram comparados aos obtidos por métodos clássicos da literatura: máquinas de vetores de suporte (SVM) [Cortes e Vapnik 1995], floresta aleatória (RF) [Breiman 1996], *naive* Bayes [Lewis e Ringuette 1994], utilizando sua variação Gaussiana, que é propícia para dados contínuos (NB) e k -vizinhos mais próximos (kNN) [Salton e McGill 1986].

O GMDL foi implementado totalmente na linguagem de programação C++. Os demais métodos foram implementados na linguagem de programação Python usando funções disponíveis na biblioteca `Scikit-learn`².

Como o desempenho dos métodos SVM, RF, kNN e GMDL pode ser afetado pelas escolhas dos seus hiper-parâmetros, foram realizadas buscas em grade para encontrar os melhores valores possíveis. Elas foram realizadas na partição de treinamento, usando validação cruzada *5-fold* estratificada. Nesse processo, a macro F-medida foi utilizada

²Scikit-learn, disponível em <https://scikit-learn.org>. Acessado em 04/07/2017.

para avaliar o desempenho dos métodos. Os intervalos de valores avaliados na busca em grade foram os seguintes:

- SVM: custo $\in \{0,5; 1; 3\}$, *kernel* $\in \{‘rbf’; ‘linear’\}$ e $\gamma \in \{0,1; 1; \frac{1}{n}\}$;
- RF: número de árvores $\in \{5; 10; 20; 30\}$ e *criterion* (função usada para avaliar a qualidade de uma determinada divisão dos dados) $\in \{‘gini’; ‘entropy’\}$;
- kNN: número de vizinhos $\in \{3; 5; 7; 9; 11\}$;
- GMDL: $\sigma^2 \in \{0,5; 1; 2; 3\}$ e $\Omega \in \{8; 16; 32; 64\}$.

Utilizou-se o fator de esquecimento unitário para o GMDL, tendo em vista que o cenário dos experimentos não é incremental.

5. Resultados

A Tabela 2 apresenta os valores de macro e micro F-medida obtidos por cada método. Os valores destacados em negrito indicam os melhores resultados por base de dados.

Tabela 2. Resultados obtidos na avaliação experimental.

Método	Macro F	Micro F	Método	Macro F	Micro F
BreastCancer			Skin		
GMDL	0,95	0,96	GMDL	0,92	0,94
kNN	0,94	0,95	kNN	1,00	1,00
NB	0,92	0,93	NB	0,87	0,92
RF	0,95	0,96	RF	1,00	1,00
SVM	0,92	0,93	SVM	0,99	0,99
CoverType			Wine		
GMDL	0,47	0,48	GMDL	1,00	1,00
kNN	0,75	0,84	kNN	0,95	0,94
NB	0,45	0,63	NB	0,97	0,97
RF	0,91	0,95	RF	1,00	1,00
SVM	0,78	0,85	SVM	0,97	0,97
Iris			WineRed		
GMDL	0,94	0,93	GMDL	0,25	0,39
kNN	0,97	0,97	kNN	0,29	0,55
NB	0,93	0,93	NB	0,31	0,54
RF	0,93	0,93	RF	0,32	0,65
SVM	0,97	0,97	SVM	0,32	0,60
Letter			WineWhite		
GMDL	0,69	0,68	GMDL	0,26	0,34
kNN	0,91	0,91	kNN	0,28	0,53
NB	0,64	0,64	NB	0,23	0,41
RF	0,95	0,95	RF	0,39	0,66
SVM	0,96	0,96	SVM	0,27	0,58

De modo geral, observa-se que o protótipo do método proposto foi marginalmente superior ao NB na maioria das bases de dados. É notório também que o método RF obteve quase sempre os melhores resultados. Porém, o GMDL obteve desempenho equiparável ao método RF na base de dados *Wine*, atingindo o melhor valor possível de macro e micro F-medida.

O GMDL obteve resultados inferiores à maioria dos métodos nas bases de dados altamente desbalanceadas como a *CoverType* e *WineRed*. O NB também obteve resultados ruins nessas duas bases de dados. Isso indica que métodos probabilísticos como o GMDL e NB podem ser mais afetados pelo problema de desbalanceamento do que métodos baseados em outras estratégias de seleção de modelo.

Tabela 3. Ranking dos métodos avaliados com base na macro F-medida.

Base de dados	GMDL	kNN	NB	RF	SVM
BreastCancer	1,5	3	4,5	1,5	4,5
CoverType	4	3	5	1	2
Iris	3	1,5	4,5	4,5	1,5
Letter	4	3	5	2	1
Skin	4,5	1,5	4,5	1,5	3
Wine	1,5	5	3,5	1,5	3,5
WineRed	5	4	3	1,5	1,5
WineWhite	4	2	5	1	3
Rank total	23	27,5	35	14,5	29
Rank médio	2,88	3,44	4,38	1,81	3,63

Para avaliar de forma mais precisa o desempenho geral dos métodos, foi realizado o teste não paramétrico de Friedman, usando as macro F-medidas obtidas pelos métodos e seguindo a metodologia apresentada em [Zar 2009, Seção 12.7]. A Tabela 3 apresenta os *rankings* dos métodos avaliados com base na macro F-medida. É importante destacar que apesar de ainda ser apenas um protótipo, o desempenho geral do GMDL apresentou *ranking* médio melhor que o de métodos estabelecidos, tais como SVM, kNN e NB.

O teste de Friedman verifica se é possível ou não rejeitar a hipótese nula, que afirma que os métodos possuem desempenhos equivalentes. Como foram usados 5 métodos e 8 bases de dados, para um intervalo de confiança $\alpha = 0,05$, o valor crítico na distribuição chi-quadrado é 9,3. Dado que o valor crítico calculado pelo teste de Friedman foi 9,11, a hipótese nula não pôde ser rejeitada e, portanto, é possível afirmar que os métodos foram estatisticamente equivalentes para as bases de dados utilizadas. Essa equivalência aos métodos tradicionais evidencia a capacidade do GMDL e o quão promissor ele pode ser, uma vez que muitas melhorias ainda podem ser efetuadas. Além disso, é importante destacar que o GMDL possui características altamente desejáveis em métodos de classificação, como ser naturalmente incremental, multiclasse e robusto ao sobreajustamento aos dados.

6. Conclusão

Neste trabalho, foi apresentado o protótipo de um método de classificação baseado no princípio MDL. Este método, nomeado GMDL, utiliza estimativas de densidade para cada um dos atributos das amostras em uma mistura Gaussiana e emprega a densidade em um ponto na mistura para fazer a predição da classe de amostras não-rotuladas. O método provê naturalmente uma troca benéfica entre a acurácia e a complexidade do modelo, uma característica oriunda da natureza da navalha de Occam e formalizada pelo princípio MDL, o que o torna robusto ao conhecido problema de sobreajustamento aos dados.

O método apresentado foi criado para ser genérico, isto é, poder ser aplicado em qualquer problema de classificação que possa ser representado por atributos categóricos ou numéricos. Ele ainda pode ser naturalmente aplicado em problemas multiclasse, sem a necessidade de utilização de técnicas de decomposição dos problemas em vários problemas binários. Ainda, ele é eficiente e gera seu modelo de predição de maneira incremental, o que o torna escalável e apto a ser aplicado em problemas de larga escala.

O desempenho do GMDL foi avaliado em um cenário de aprendizado em *batch*, usando bases de dados clássicas da literatura e comparado aos resultados obtidos por métodos considerados referência na área: NB, KNN, SVM e RF. O método proposto obteve o segundo melhor *ranking* médio e a análise estatística dos resultados evidenciou que o GMDL foi equivalente a todos os métodos usados na comparação. Esse resultado indica

que o GMDL, apesar de ainda estar em fase inicial de criação, já pode apresentar resultados no mesmo nível de métodos clássicos, mas podendo funcionar de maneira totalmente incremental e sem a necessidade de ajuste inicial nos dados.

Em trabalhos futuros, serão adotadas novas estratégias para tentar melhorar o poder preditivo do método. Por exemplo, uma função de atribuição de pesos aos atributos de acordo com seu poder preditivo poderá ser acoplada à função principal do GMDL. Também, está prevista a condução de experimentos no cenário de classificação *online* utilizando os dados sem qualquer tipo de tratamento. Ainda, pretende-se avaliar o GMDL em cenários onde os dados são esparsos e possuem alta dimensão, usando um número maior de bases de dados e explorando mais extensivamente os meta-parâmetros de cada método.

Agradecimentos

Os autores são gratos à agência de fomento CAPES (Proc. 1709642) pelo apoio financeiro a este projeto de pesquisa.

Referências

- Barron, A. e Cover, T. (1991). Minimum complexity density estimation. *IEEE Transactions on Information Theory*, 37(4):1034–1054.
- Barron, A., Rissanen, J., e Yu, B. (1998). The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760.
- Braga, I. A. e Ladeira, M. (2008). Filtragem adaptativa de spam com o princípio minimum description length. In *Anais do XXVIII Congresso da Sociedade Brasileira de Computação (SBC'08)*, pages 11–20, Belém, Brasil. Sociedade Brasileira de Computação (SBC).
- Bratko, A., Cormack, G. V., Filipič, B., Lynam, T. R., e Zupan, B. (2006). Spam Filtering Using Statistical Data Compression Models. *Journal of Machine Learning Research*, 7(12):2673–2698.
- Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.
- Cortes, C. e Vapnik, V. N. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.
- Ferreira, J., Matos, D. M., e Ribeiro, R. (2016). Fast and Extensible Online Multivariate Kernel Density Estimation. *CoRR*, abs/1606.0(1):1–17.
- Friedman, N., Geiger, D., e Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning*, 29(2):131–163.
- Grünwald, P. (2000). Model Selection Based on Minimum Description Length. *Journal of mathematical psychology*, 44:133–152.
- Grünwald, P. (2005). A tutorial introduction to the minimum description length principle. *Advances in minimum description length: Theory and applications*, 1(1):23–81.
- Hansen, M. H. e Yu, B. (2001). Model Selection and the Principle of Minimum Description Length. *Journal of the American Statistical Association*, 96(454):746–774.
- Hoi, S. C. H., Wang, J., e Zhao, P. (2014). Libol: A library for online learning algorithms. *Journal of Machine Learning Research*, 15(1):495–499.
- Kolmogorov, A. N. (2016). On Tables of Random Numbers. *Sankhya: The Indian Journal of Statistics, Series A*, 53(4):369–376.

- Kononenko, I. (1998). The minimum description length based decision tree pruning. In *Pacific Rim International Conference on Artificial Intelligence (PRICAI'98)*, pages 228–237, Singapura, Singapura. Springer, Springer.
- Kristan, M., Leonardis, A., e Skočaj, D. (2011). Multivariate online kernel density estimation with Gaussian kernels. *Pattern Recognition*, 44(10-11):2630–2642.
- Lam, W. e Bacchus, F. (1994). Learning Bayesian Belief Networks: An Approach Based on the Mdl Principle. *Computational Intelligence*, 10(3):269–293.
- Lewis, D. D. e Ringuette, M. (1994). A Comparison of Two Learning Algorithms for Text Categorization. In *3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, pages 81–93, Las Vegas, NV, EUA. Information Science Research Institute, University of Nevada.
- Lichman, M. (2013). UCI Machine Learning Repository.
- Ling, R. F. (1974). Comparison of Several Algorithms for Computing Sample Means and Variances. *Journal of the American Statistical Association*, 69(348):859.
- Lüthke, J. (2013). *Location Prediction Based on Mobility Patterns in Location Histories*. M.sc. thesis, Hamburg University of Technology.
- MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*, volume 4. Cambridge University Press.
- McLachlan, G. e Peel, D. (2004). *Finite mixture models*, volume 1. John Wiley & Sons, Inc.
- Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465–471.
- Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431.
- Ross Quinlan, J. e Rivest, R. L. (1989). Inferring decision trees using the minimum description length principle. *Information and Computation*, 80(3):227–248.
- Salton, G. e McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.
- Silva, R. M., Almeida, T. A., e Yamakami, A. (2017). MDLText: An efficient and lightweight text classifier. *Knowledge-Based Systems*, 118:152–164.
- Silverman, B. W. (1986). *Density estimation for statistics and data analysis*, volume 26. CRC Press, 1 edition.
- Tony F. Chan, Gene H. Golub, R. J. L. (1983). Algorithms for Computing the Sample Variance: Analysis and Recommendations. *The American Statistician*, 37(3):242–247.
- Wand, M. P. e Jones, M. C. (1994). *Kernel smoothing*, volume 1. Crc Press, 1 edition.
- Welford, B. P. (1962). Note on a Method for Calculating Corrected Sums of Squares and Products. *Technometrics*, 4(3):419.
- Zar, J. H. (2009). *Biostatistical Analysis*. Prentice Hall, 5 edition.
- Zivkovic, Z. e van der Heijden, F. (2004). Recursive unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):651–6.

Denoising Auto-Encoders as Feature Extractors in Hierarchical Classification Problems

Felipe Kenji Nakano, Ricardo Cerri

¹Department of Computer Science - Federal University of São Carlos
Rodovia Washington Luís, km 235 - São Carlos - SP - Brazil
Email: felipe.nakano@dc.ufscar.br / cerri@dc.ufscar.br

Abstract. *Real world datasets are likely to be huge and noisy. In order to correctly handle such high dimensional datasets, Dimensionality Reduction techniques are investigated. These techniques aim to obtain compact representations of data. Many works have addressed Dimensionality Reduction techniques in traditional classification problems, however fewer studies have been performed for Hierarchical Classification, where classes assigned to an instance are organized in a hierarchy. Recently, many fields of study have expanded their state-of-art by employing Denoising Auto-Encoders (DAEs), which are neural networks capable of learning a codification of their input data. In this work, we investigate how DAEs behave as feature extractors for Hierarchical Classification scenarios, using protein function prediction datasets as a case study. Our results show that DAEs can extract more compact representations while keeping classifiers performances.*

1. Introduction

Real world datasets are likely to be noisy and have high dimensionality. Such properties are undesired since it may hinder classifiers' capacities, making the task of classification harder. In order to correctly handle high amounts of data, dimensionality reduction techniques have been studied [Van Der Maaten et al. 2009].

Dimensionality Reduction (DR) can be defined as the process of obtaining more compact presentations of the original data, capturing information required for higher level decision making [Saul and Roweis 2003]. By reducing the dimensionality, undesired properties such as useless information and noisy features should be removed [Tang et al. 2014].

In general, DR algorithms can be split in two groups: Feature Selection (FS) and Feature Extraction (FE). The latter creates a new representation of the data, i.e., the original features are combined in order to create new ones. Whereas FS consists of reducing the original features to a subset of them [Tang et al. 2014]. Even though DR has been vastly addressed in traditional classification, fewer works have explored it in the Hierarchical Classification (HC) context. In HC problems, classes are structured in a hierarchical fashion. Such hierarchy defines relationships between superclasses and subclasses. Likewise traditional classification, HC suffers from high dimensional and noisy representations.

The literature of DR in HC can be misleading. Works like [Masci et al. 2011] have been explicitly referred as hierarchical feature learning, however these methods extract features using hierarchical methods for traditional classification problems. Furthermore,

in Freitas' review [Wan and Freitas 2017], hierarchical feature learning is used to address feature selection in hierarchically structured features spaces.

Differently from these previous works, we are dealing with DR in HC scenarios, considering classes structured in a hierarchy. Similarly to traditional classification, DR methods for HC also try to obtain subsets of features with high representational power. Nonetheless, hierarchical constraints should be considered. A simple way of coupling with this consists of applying DR methods in a local fashion, where different features are extracted and used for each node of the hierarchy.

Recently, Auto-Encoders (AE) have been explored as a DR method, more specifically, variations such as Denoising Auto-Encoder (DAE). Since the work of Hinton [Hinton and Salakhutdinov 2006] and [Vincent et al. 2008], DAEs have shown potential in many fields of study. DAEs are feed-forward neural networks whose hidden layer learns a codification of the input data with random corruption. Such codification corresponds to a new representation of the data and can be used for classification. According to [Wang et al. 2016], AEs, in general, can approximate the intrinsic dimensionality of the data, producing different features than other algorithms.

As reported in [Fukunaga 2013], the intrinsic dimensionality of data corresponds to the minimal amount of features necessary to capture its properties. Thus, DAEs can reduce the dimensionality of the data by creating a smaller representation of equal or better representational power. In this work, we present how DAEs behave as feature extractors in the HC context, using protein function prediction datasets as a case study. By training a DAE per parent node, and using its codification as features, our results show that DAEs are able to extract meaningful features, and successfully produce compact representations.

The remainder of this paper is organized as follows. Section 2 brings recent works. Section 3 defines the HC problem along with approaches to solve it. Section 4 presents AEs and their variant Denoising Auto-Encoder (DAE). Section 5 details our materials, methods, datasets and evaluation measures. Next, Section 6 contains our results and discussion. Finally, Section 7 presents our findings, as well as future works.

2. Related Works

As already stated, some works may mislead the reader regarding DR for HC problems. Although their titles clearly point to DR in HC, these works investigate other fields of study. In the work of [Masci et al. 2011], Convolution Auto-Encoders are stacked to extract features for traditional classification in a hierarchical way.

Another field of study, which is called hierarchical features selection, is surveyed in [Wan and Freitas 2017]. These works address hierarchically structured feature spaces where dependencies are exploited to perform feature selection. However, there is no class hierarchy in the problem.

Recall that, although mentioning hierarchical feature selection, none of the previously mentioned works perform DR considering hierarchical scenarios. Following, we present some works proposed for scenarios with hierarchically structured classes.

2.1. Dimensionality Reduction for Hierarchical Classification

Differently from traditional classification, DR in HC should be able to handle hierarchical constraints. This can be achieved in a global way where a fixed amount features is extracted and used for the entire hierarchy [Slavkov et al. 2013], or locally by sampling many subsets of features with possible different dimensions [Nemati et al. 2009, Kamal et al. 2015, Naik and Rangwala 2016]. The following works presents some DR methods for HC.

An hybrid approach that combines Ant Colony Optimization and Genetic Algorithms (GAs) is proposed in [Nemati et al. 2009]. Their results show that, when combined, both algorithms are able to select a subset of features with superior results.

The work of Kamal [Kamal et al. 2015] also employed (GAs). First, the Correlation Feature Selection algorithm is applied. Next, a Wrapper method is added. For each local classifiers, a optimized subset of features is used. This DR method did not improve their results, but reduced the processing time.

In the work of Naik [Naik and Rangwala 2016], the filter-based feature selection methods, Gini-index, Minimal Redundancy Maximal Relevance (MRMR) and Kruskal-Wallis (KW), are studied. Both MRMR and KW are applied once for the entire hierarchy. The Gini-index is also used globally. Besides that, the authors proposed an local strategy that applies the Gini-Index in local classifiers. Even though their results did not improve, the selected features maintained very close results to the original features.

The work of Slavkov [Slavkov et al. 2013] proposes an extension of the algorithm Relief for hierarchical feature selection, called HMC-Relief. By applying Weight Euclidean Distance, the Bayes rule and relationships established by the hierarchy, this algorithm is able to rank features. Their results show that HMC-Relief can select a subset of features with higher discriminatory capacities than random features.

Recall that although considering hierarchical scenarios, the previous works perform feature selection, while we perform feature extraction. Also, some works are proposed for multi-label classification, where two or more paths in the hierarchy can be assigned to an instance. In our proposal, we consider that an instance can be classified into an unique hierarchical path.

3. Hierarchical Classification

Traditional classification problems consist of training a model able to predict the class of a new instance. In Hierarchical classification (HC), the classes are organized according to a taxonomy (hierarchy) which defines relationships among subclasses and superclasses.

Up to the moment, two types of taxonomies are known in the literature, Tree and DAG (directed acyclic graph). Trees allow only one superclass per node, while DAGs permit many superclasses per node. Intuitively, DAG problems are likely to be more complex.

As stated in [Silla and Freitas 2010], HC problems are defined by a tuple (γ, ψ, ϕ) , where γ describes the taxonomy (Tree or DAG), ψ describes whether the instances are classified into multiple paths or into a single path to a node, and ϕ stands for non-mandatory leaf-node classification, or mandatory leaf-node classification, i.e., if internal nodes can or cannot be the deepest predicted classes.

Two main approaches for HC are available in the literature, global and local. The local approach reduces the problem to multiple smaller problems using a hierarchy of out-of-shelf classifiers, like neural networks and decision trees. Different reduction strategies are used in the local approach, LCL (**L**ocal **C**lassifier per **L**evel) trains a classifier per level (depth) of the taxonomy [Cerri et al. 2016]. LCPN (**L**ocal **C**lassifier per **P**arent **N**ode) trains a classifier for each parent node [Kiritchenko 2004], while LCN (**L**ocal **C**lassifier per **N**ode) trains a binary classifier for each node [Valentini and Re 2009].

In all strategies, the outputs of classifiers are combined to classify a new instance. The Local approach is more demanding in terms of processing and is prone to error propagation. However, it is intuitive, simpler to implement, and allow the power of any classifier to be exploited.

The global approach trains a single model for the whole hierarchy. Traditional classifiers cannot be used, since taxonomy related aspects are taken for granted. Thus, many global-based methods employ adaptations.

4. Auto-Encoders

In the work of Hinton [Hinton and Salakhutdinov 2006], Auto-Encoders (AEs) were proposed as dimensionality reducers. AEs are two layer feed-forward neural networks whose hidden layer learns a codification of the original data [Liu et al. 2016].

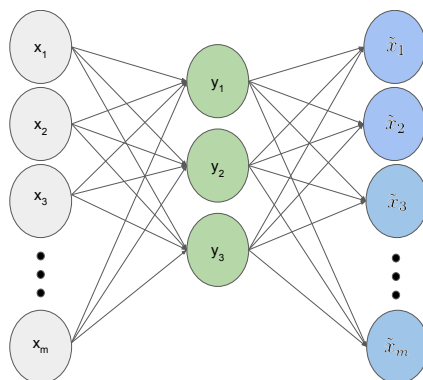


Figure 1. Auto-encoders architecture. Green and blue nodes corresponds respectively to the new representation learned, and the reconstructed input.

As seen in Figure 1, the architecture of AEs resembles MultiLayer Perceptrons, but the output layer has the same dimensionality as the initial layer. This is necessary because AEs encode the original data (x) and reconstruct it back to its original values (\tilde{x}). The hidden layer outputs (y) consists of a codification of the original data. Thus AEs are able to extract features, and can be used for DR and information compression [Hinton and Salakhutdinov 2006].

The training of AEs is usually done using the Backpropagation algorithm [Rumelhart et al. 1986]. Recall that the target values (final layer outputs) are the own inputs. One key advantage of AEs is that, during their training, meaningful features are continuously extracted, and useless information is filtered [Liu et al. 2016].

AEs may not be robust in some situations, especially noisy datasets. Due to this limitation, the work of Vincent [Vincent et al. 2008] introduced a new type of AE called

Denosing Auto-Encoder (DAE). They extend the original concept of AE to enforce robustness. This is achieved with random corruption of the original data. Such random noise insertion avoids trivial solutions. In its training phase, a DAE partially corrupts its input data by turning some values to 0, and then reconstructs then back to the original values (no corruption).

5. Materials and Methods

In this work, we investigate how DAEs' feature extraction capacities perform on a hierarchical classification scenario. For classification, we employed two variations of the Local Classifier per Parent Node (LCPN) strategy proposed in [Nakano et al. 2017] along with DAEs as feature extractors. Thus, one DAE is associated to each parent node, and its extracted representation is directly used as features. We have chosen these variations since they can output non-mandatory leaf-node predictions.

In all experiments, we employed the KNN algorithm as the base classifier for local strategies, with $k = 3$. The KNN was chosen since it requires almost no parameter tuning.

Following, we present details on the local hierarchical classification strategies used, and the DAEs implementation methods. We also detail the evaluation measures and datasets employed.

5.1. Hierarchical Classification Methods

In a recent work, two new local classification strategies were introduced [Nakano et al. 2017]. These strategies are modifications of the LCPN strategy [Kiritchenko 2004], and were proposed specifically to cope with non-leaf mandatory classifications.

5.1.1. LCPN

As the name suggest, LCPN (**l**ocal **c**lassifier per **p**arent **n**ode), trains a classifier per parent node, able to distinguish between its children nodes. Taking Figure 2 as an example, one classifier is trained for each dashed node (*Root*, 1, 1.1, 2.1.1). Note that the nodes 2 and 2.1, are parent nodes, however they have only one child node. Thus, there is no need to associate a classifier to distinguish between their children.

In the classification phase, a baseline method is used. At first, the instance is classified by the *Root* node classifier, then the classifier belonging to the node predicted is used until a leaf node is reached.

As seen in Figure 2, the node 2.1 contains only one child node (2.1.1), hence there is no need to associate a classifier to node 2.1, since there is 100% of probability that this classifier will choose class 2.1.1. Thus, applying a threshold of 1.0 (100% of chance) would remove all classifications. Because of this, [Nakano et al. 2017] proposed two strategies to naturally cope with non-leaf classification, nLLCPN and LCPNB, aiming at minimizing the error propagation.

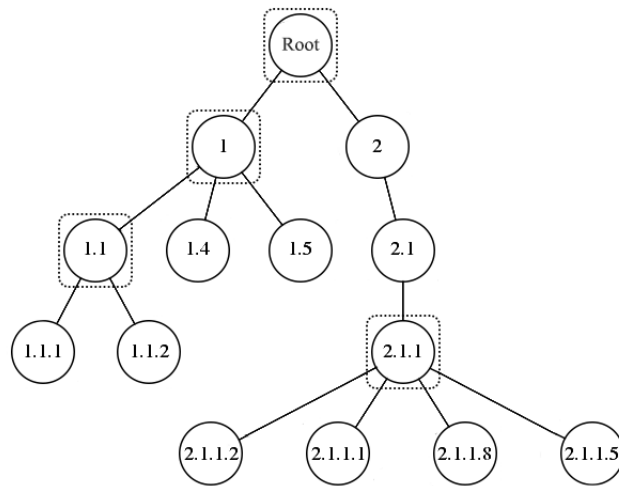


Figure 2. LCPN training and classification strategy.

5.1.2. nLLCPN

Using thresholds can be troublesome. First, defining a threshold can be challenging since different datasets may demand different values. Second, many classifiers, like Decision Trees and Naive Bayes, output biased probabilities. Due to these flaws, the approach nLLCPN (**n**on-**L**ea**f** **L**ocal **C**lassifier per **P**arent **N**ode) automatically enables non-leaf classification by modifying the problem's taxonomy. Figure 3 presents an example of this modification. Grey nodes represent the modifications performed.

Taking Figure 3 as an example, parent classifiers 1.1 and 2.1 are trained to distinguish between their children nodes and themselves. As positive instances for the grey nodes, only instances with the exact label are considered, i.e., only instances having 2.1 or 1.1 as their deepest classes.

In the classification phase, a top-down approach just like LCPN's is used. However, if a node predicts itself (grey nodes), the classification phase ends and a non-leaf classification is obtained.

5.1.3. LCPNB

Another strategy, LCPNB (**L**ocal **C**lassifier per **P**arent **N**ode and **B**ranch), uses the same training procedure as nLLCPN, i.e., the taxonomy is also slightly modified. Its difference comes in the classification phase. As specified in LCPN, the top-down classification is prone to error propagation. The rationale behind LCPNB is that errors occur when the classifier outputs similar probabilities for different classes. Thus, instead of expanding the node with highest probability (LCPN and nLLCPN strategies), LCPNB uses all parent classifiers to produce the final classification.

More specifically, LCPNB evaluates classification probabilities considering all paths. For all parent classifiers in the taxonomy, starting from the *Root*, classification probabilities to all possible nodes are summed up. Then, the average probability is calcu-

lated, and the path with the highest probability is considered the final classification. As an example, consider classifiers producing probabilities depicted in Figure 3. The LCPNB strategy would classify an instance into class 2.1, since it presents the highest average probability, given by

- $2(0.4) \rightarrow 2.1(1.0) \rightarrow 2.1(0.8)$
average = $(0.4 + 1.0 + 0.8)/3 = 0.73$.

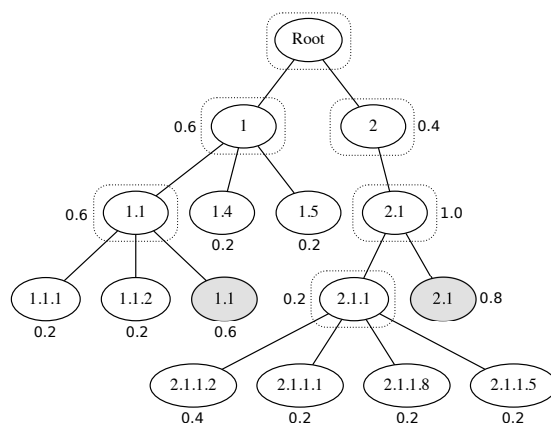


Figure 3. nLLCPN and LCPNB phases. Adapted from [Nakano et al. 2017]

5.2. Datasets

In this work, we used G protein-coupled receptors (GPCR) and Enzyme datasets. GPCR are a family of proteins able to detect molecules outside of a cell and react to them [Trzaskowski et al. 2012]. Enzymes are molecules, mostly proteins, that accelerate chemical reactions in general [Placzek et al. 2017]. In both datasets, proteins are represented by unique hierarchical indexes, by GPCRDB codes in the case of GPCRs, and Enzyme Commission Codes (EC) in the case of enzymes [Costa et al. 2007].

As features, for both GPCR and EC datasets, protein signatures (motifs), the sequence length and the molecular weight are used. Motifs are binary features which are built by checking the presence of motifs in four public databases (Interpro, Pfam, Prosite and Prints), i.e. 0 for absence and 1 for presence. The sequence length presents the count of amino acids a sequence has, and molecular weight corresponds to the weight of each amino acid summed up. Table 1 presents the characteristics of the datasets used.

5.3. Evaluation Measures

As recommended by Silla and Freitas [Silla and Freitas 2010], we employed the hierarchical classification specific evaluation measures *hierarchical precision (hP)*, and *hierarchical recall (hR)*. In these equations, C_i and Z_i correspond, respectively, to the set of true and predicted classes for a given instance i . We also calculated the *hierarchical f-measure (hF)*, corresponding to the harmonic mean of hP and hR.

Table 1. Datasets statistics

Dataset	# Instances	# Features	# Classes Per Level
ECInterpro	13902	1216	6 / 41 / 96 / 187
ECProsite	13877	585	6 / 42 / 89 / 187
ECPrints	13828	382	6 / 45 / 92 / 208
ECPfam	13871	708	6 / 41 / 96 / 191
GPCRInterpro	7353	450	12 / 54 / 82 / 50
GPCRProsite	6167	129	9 / 50 / 79 / 49
GPCRPrints	5324	283	8 / 46 / 76 / 49
GPCRPfam	6967	75	12 / 52 / 79 / 49

$$hP = \frac{\sum_i |Z_i \cap C_i|}{\sum_i |Z_i|} \quad (1) \quad hR = \frac{\sum_i |Z_i \cap C_i|}{\sum_i |C_i|} \quad (2) \quad hF = \frac{2 * hP * hR}{hP + hR} \quad (3)$$

5.4. Statistical Test

In order to verify the superiority of the methods using the procedure suggested in [Demšar 2006]. At first, Friedman’s test, with significance level at $\alpha = 0.05$, is used verify if any method is significantly different from others (null hypothesis rejection). Next, post hoc Nemenyi test is applied, basically, forms groups of data that differ statistically, according to a Critical Difference (CD) value. We used the hF measure values for the tests.

5.5. Proposal: Denoising Autoencoders as Feature Extractors for HC

Considering that to the best of the authors’ knowledge, this is the first work to investigate DAEs as feature extractors in hierarchical datasets, we have decided to explore simple architectures, with one single hidden layer having different amounts of hidden neurons (features produced).

Since we are employing LCPN based strategies, for each parent node with two or more children, a DAE is trained and its codification learned is used as features for the base classifier. Taking Figure 2 as an example, a DAEs is trained for the nodes $\{Root, 1, 1.1, 2.1.1\}$. Recall that if other Local strategies were employed, DAEs would be associated in a different way, level-wise in the case of LCL or node-wise for LCN.

Being $|N|$ the original number of features, we tested hidden layer sizes ranging from 10% up to 90% of $|N|$. Thus, supposing 10% is used, the hidden layer size is given by $|N| \times 0.1$. In this work, we maintained a fixed amount of hidden layer neurons (features) for all DAEs throughout the hierarchy, since exploring DAEs with different dimensionalities may result in a exponential number of possibilities. The activation function used was the Rectified Linear Unit (ReLU).

6. Experiments

We used the 10-fold cross-validation strategy, and reported the average of the measures over the 10 executions. We also normalized the features, making their values in the interval $[0, 1]$. Besides that, we removed classes with less than 10 examples.

For empirical comparison, we used Principal Component Analysis (PCA) as a DR method. We decide to use PCA, since it generates new features based on the original ones, just like DAEs. Thus, for each parent node, PCA is used locally.

Table 2 presents our results. Each column represents the hidden layer sizes as a percentage of the number of features, starting from 10% up to 90%. The same applies for the number of components generated by PCA. The last column of each dataset presents the results using the original normalized features.

Our results showed that DAEs can be effective as feature extractors in HC problems. In EC and GPCR datasets, all experiments show that DAEs can indeed produce results very close to the original data. The same applies for PCA, however, in the situations where a small amount of features is used, DAEs had the upper hand.

We can also verify that the datasets contain many meaningless features. By analyzing then, we can notice they are vastly sparse, i.e., motifs are more likely to be absent than present. In such scenarios, DAEs were able to extract very meaningful representations. Even in minimal situations, like GPCRpfam and GPCRProsites, 10% of features (7 and 12 features respectively) were sufficient to yield competitive results against raw data.

These results are supported by statistical tests presented in Figures 4 and 5. Using the nLLCPN strategy, there is no statistical difference among Raw, DAE's and PCA with 90%, 70% and 50% of components, nonetheless DAE with 10% is superior than PCA.

In the case of LCPNB, a similar behaviour was noted. The test did not point statistical difference between DAE, PCA (90%, 70%, 50%), and Raw data. PCA with smaller amounts of components and DAE with 10% did performed worse.

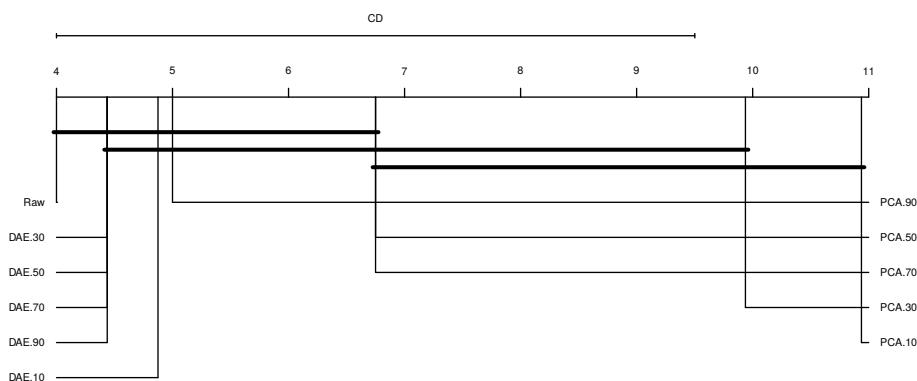


Figure 4. Nemenyi Test for nLLCPN

7. Conclusion

In this work, we evaluated how DAEs behave as local feature extractors in HC problems using LCPN based strategies, considering a case study with protein function datasets. By using features extracted by DAEs with, our results showed that DAEs can extract meaningful features in HC problems.

Taking a look at the GPCR and EC datasets, we can notice that DAEs behaved positively. In most of the experiments, no significant deterioration was found. Even a minimal amount of features is sufficient to produce competitive results. We can also notice that as the number of features extracted increases, the results also slightly raises.

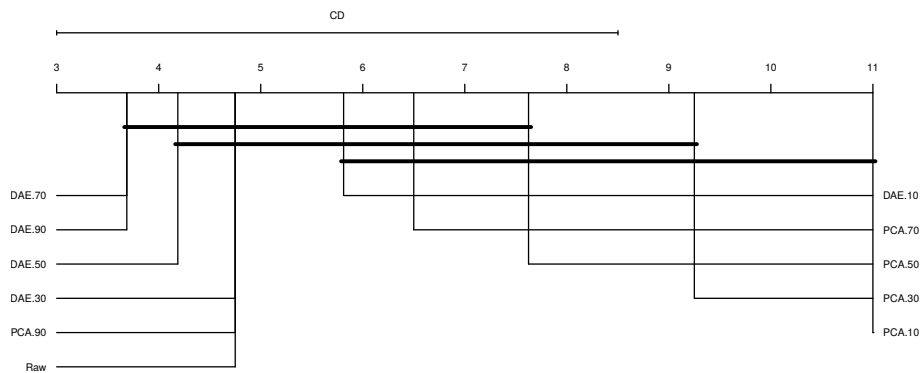


Figure 5. Nemenyi Test for LCPNB

However, using features extracted by DAEs did not improve the results in some cases. In future works, different amounts of features for each local classifier, along with other neural networks with feature extraction capacities should be explored.

Acknowledgements

The authors would like to thank FAPESP-SP, CAPES and CNPq. Felipe Kenji Nakano is funded by grant 2016/12489-2. CERRI COLOCAR SEUS PROJETO AKI

References

- Cerri, R., Barros, R. C., P. L. F. de Carvalho, A. C., and Jin, Y. (2016). Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC Bioinformatics*, 17(1):373.
- Costa, E. P., Lorena, A. C., Carvalho, A. C. P. L. F., Freitas, A. A., and Holden, N. (2007). Comparing several approaches for hierarchical classification of proteins with decision trees. In *Proceedings of the 2Nd Brazilian Conference on Advances in Bioinformatics and Computational Biology*, pages 126–137, Berlin, Heidelberg. Springer-Verlag.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30.
- Fukunaga, K. (2013). *Introduction to statistical pattern recognition*. Academic press.
- Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.
- Kamal, N. A. M., Bakar, A. A., and Zainudin, S. (2015). Filter-wrapper approach to feature selection of gpcr protein. In *Electrical Engineering and Informatics (ICEEI), 2015 International Conference on*, pages 693–698. IEEE.
- Kiritchenko, S. (2004). Famili, hierarchical text categorization as a tool of associating genes with gene ontology codes. In *DMTM-ECML/PKDD*, pages 26–30.
- Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., and Alsaadi, F. E. (2016). A survey of deep neural network architectures and their applications. *Neurocomputing*.
- Masci, J., Meier, U., Cireşan, D., and Schmidhuber, J. (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. In *Proceedings of the 21th International*

- Conference on Artificial Neural Networks - Volume Part I, ICANN'11*, pages 52–59, Berlin, Heidelberg. Springer-Verlag.
- Naik, A. and Rangwala, H. (2016). Embedding feature selection for large-scale hierarchical classification. In *IEEE International Conference on Big Data*, pages 1212–1221.
- Nakano, F. K., Pinto, W. J., Pappa, G. L., and Cerri, R. (2017). Top-down strategies for hierarchical classification of transposable elements with neural networks. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2539–2546.
- Nemati, S., Basiri, M. E., Ghasem-Aghaee, N., and Aghdam, M. H. (2009). A novel aco-ga hybrid algorithm for feature selection in protein function prediction. *Expert Systems with Applications*, 36(10):12086 – 12094.
- Placzek, S., Schomburg, I., Chang, A., Jeske, L., Ulbrich, M., Tillack, J., and Schomburg, D. (2017). Brenda in 2017: new perspectives and new tools in brenda. *Nucleic Acids Research*, 45(D1):D380–D388.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, USA.
- Saul, L. K. and Roweis, S. T. (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4(Jun):119–155.
- Silla, Carlos N., J. and Freitas, A. (2010). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72.
- Slavkov, I., Karcheska, J., Kocev, D., Kalajdziski, S., and Dzeroski, S. (2013). Extending relief for hierarchical multi-label classification. *machine learning*, 4:13.
- Tang, J., Alelyani, S., and Liu, H. (2014). Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, page 37.
- Trzaskowski, B., Latek, D., Yuan, S., Ghoshdastider, U., Debinski, A., and Filipek, S. (2012). Action of molecular switches in gpcrs-theoretical and experimental studies. *Current medicinal chemistry*, 19(8):1090–1109.
- Valentini, G. and Re, M. (2009). Weighted true path rule: a multilabel hierarchical algorithm for gene function prediction. In *MLD-ECML/PKDD*, pages 132–145.
- Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10:66–71.
- Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.
- Wan, C. and Freitas, A. A. (2017). An empirical evaluation of hierarchical feature selection methods for classification in bioinformatics datasets with gene ontology-based features. *Artificial Intelligence Review*, pages 1–40.
- Wang, Y., Yao, H., and Zhao, S. (2016). Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242.

Extração de tópicos de interesse de pesquisas brasileiras em Inteligência Artificial

Marcelo K. Albertini¹, Adriano L. de Sá¹

¹Faculdade de Computação – Universidade Federal de Uberlândia (UFU)
Caixa Postal 593 – 38408-100 – Uberlândia – MG – Brazil

{adrianolima, albertini}@ufu.br

Abstract. *The research field in Artificial intelligence is remarkable for its long history and for having passed through various changes of interest over the years. In this article, the evolution of the productivity and the research topics of Brazilian researchers in this field was studied. With this intention, a quantitative analysis of the trajectories of researchers and their relationship with the prestige of higher education institutions is presented. Additionally, we employ the techniques classification One-Class Support Vector Machine and extracting topics Latent Dirichlet allocation to find the search topics that most characterize the publications year by year.*

Resumo. *O campo de pesquisa em Inteligência Artificial é notável por sua longa história e por ter passado por várias mudanças de interesse no decorrer dos anos. Neste artigo estudou-se a evolução da produtividade e dos tópicos de pesquisas de pesquisadores brasileiros nesse campo. Com esse intuito, apresenta-se uma análise quantitativa das trajetórias de pesquisadores e sua relação com o prestígio de instituições de ensino superior. Adicionalmente, empregamos as técnicas de classificação One-Class Support Vector Machine e extração de tópicos Latent Dirichlet Allocation para encontrar os tópicos de pesquisa que mais caracterizam as publicações ano a ano.*

1. Introdução

Publicações acadêmicas são atualmente o principal modo de disseminação de conhecimento científico. A quantidade e o conteúdo de publicações influenciam vários aspectos que permeiam a academia, tais como, contratações de novos pesquisadores, definição de políticas de governo em ciência e tecnologia e a atuação de agências de fomento de pesquisa. As principais formas de estudos sobre a produção científica concentram-se em análise de curriculum vitae, análise de citações entre publicações e caracterização de redes de grupos de colaboradores mais frequentes.

Internacionalmente, a trajetória de produtividade de pesquisadores tem sido estudada há pelo menos 60 anos. Frequentemente estudos identificam a existência de uma trajetória de produtividade canônica em que há um aumento e um declínio ao longo da carreira acadêmica do pesquisador [Cole 1979, Simonton 1997, Symonds et al. 2006].

No contexto brasileiro, estudos sobre a produtividade da pesquisa concentram-se em alguns campos de conhecimento, como Medicina [Mendes et al. 2010, Tess et al. 2009, Barata and Goldbaum 2003], Odontologia [Scarpelli et al. 2008] e Computação [Mena-Chalco et al. 2012, Digiampietri et al. 2012]. Frequentemente esses

estudos caracterizam trajetórias dos pesquisadores bolsistas de alta produtividade do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Por exemplo, Mendes et al. [Mendes et al. 2010] estudaram a produtividade de pesquisadores em Medicina. Eles analisaram as trajetórias de 38 pesquisadores bolsistas do CNPq de 2005 a 2007. Entre os pesquisadores considerados, 65,9% eram homens e 72,8% provenientes de apenas dois estados brasileiros: São Paulo e Rio de Janeiro.

Mena-Chalco et al. [Mena-Chalco et al. 2012] avaliaram o perfil das produções bibliográficas de programas de pós-graduação brasileira em computação de 2004 a 2009. Eles observaram que nessa área há uma preferência para a publicação em conferências (cerca de 71,5%), seguido por artigos em revistas (aproximadamente 15,9%) e de livros, capítulos de livros e outros (12%).

A caracterização da trajetória de pesquisa considerando somente aspectos quantitativos provê informações parciais sobre a evolução de uma área. Além disso, devido a dificuldades de coleta e processamento de dados, são escassos os estudos qualitativos sobre a evolução do conteúdo abordado em uma área.

O objetivo deste artigo foi estudar a atuação de pesquisadores brasileiros no campo da Inteligência Artificial de duas formas. A primeira foi estudar a trajetória da produtividade de pesquisadores envolvidos nessa área. Para isso buscou-se relacionar a contagem de publicações individuais com o prestígio de instituições. A segunda forma foi caracterizar a evolução dos tópicos de pesquisa. Nesse sentido, extraiu-se termos associados aos tópicos mais representativos dos artigos produzidos desde a década de 1960.

O restante deste documento está organizado da seguinte forma: na Seção 2 trata-se das fontes dos dados considerados neste estudo e como eles foram processados; na Seção 3 avalia-se as tendências das taxas de publicação de acordo com o prestígio das instituições; nas Seções 4 e 5 descreve-se as técnicas utilizadas e o processo de caracterização de tópicos de interesse; na Seção 6 apresenta-se as principais observações sobre a análise dos dados de produtividade do campo de Inteligência Artificial.

2. Dados

O Ministério da Ciência, Tecnologia e Inovação brasileiro desenvolveu e mantém a Plataforma Lattes¹, um site usado para armazenar e publicar currículos acadêmicos. Nessa plataforma cada pessoa pode divulgar quais são suas atividades mais importantes.

Durante os primeiros meses do ano de 2014, capturou-se dados dos currículos de 2967 pessoas que declararam atuar em “Ciência da Computação” e que as palavras Inteligência Artificial estavam presentes. A captura dos dados foi realizada com uma adaptação do sistema ScriptLattes a fim de realizar uma busca em nível na rede de colaborações dos pesquisadores brasileiros de modo a não sobrecarregar a Plataforma Lattes [Mena-Chalco et al. 2009]. Foram utilizados como ponto de partida os currículos dos pesquisadores de departamentos de computação de universidades do sul (UFRGS e UFSC), sudeste (USP, UFMG, UFRJ e UFU) e nordeste (UFPE). A partir desses currículos, os seguintes dados estão acessíveis: nome completo do pesquisador, detalhes de afiliação, campos de atividade e dados bibliográficos sobre artigos, livros, capítulos e

¹Plataforma Lattes: <http://lattes.cnpq.br>

outros itens de publicação. Há também dados sobre títulos acadêmicos e o ano de conclusão do doutorado com sua correspondente área (por exemplo, doutorado em Ciência da Computação ou doutorado em Engenharia Elétrica). Daqueles que declararam trabalhar em Ciência da Computação, 1449 ganharam um título de doutorado com palavras contendo prefixos de “comput” ou “inform”. Ao longo deste artigo, usamos este método para identificar os títulos na área de Computação e Informática (CI).

Uma curva dos anos de conclusão de doutorado é mostrada na Figura 1. Nessa figura, a curva rotulada com “irrestrito” inclui todos os títulos de doutorado e a curva “CI” representa a contagem apenas dos doutores em Ciência da Computação. Estas curvas mostram que a pesquisa em “CI” no Brasil iniciou seu crescimento no final década de 1980.

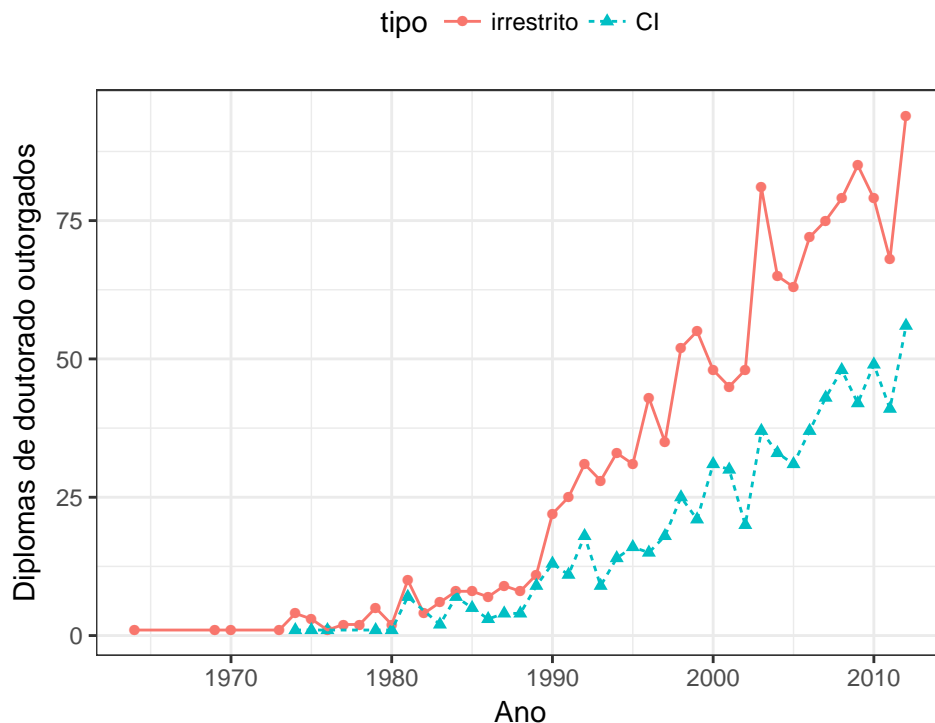


Figura 1. Crescimento do número de títulos de doutorado em Computação e Informática no final da década de 1980. Os pontos indicam o número de títulos de doutorado concedidos anualmente entre aqueles que declararam trabalhar em Ciência da Computação na área da Inteligência Artificial. A curva rotulada com “CI” indica o número de títulos de informática e “irrestrito” corresponde àqueles que trabalharam no campo da IA.

Para caracterizar o prestígio, usou-se uma lista com as 200 instituições mais bem ranqueadas de acordo com o ranking do *Webometrics* das universidades mundiais². Esse ranking usa indicadores baseados na análise de links de páginas institucionais da Web e análise de citações bibliográficas.

Neste documento, salvo indicação em contrário, as publicações contabilizadas foram dos seguintes tipos: artigos em periódicos, livros, capítulos e documentos completos

²Ranking disponível em http://www.webometrics.info/en/latin_america/brazil

em processo. Todas as publicações datam de 1965 a 2012.

Como proposto por Way et al. [Way et al. 2016], seria injusto comparar uma publicação de 1965 com uma publicação de 2012. Nas análises das Figuras 2 e 4 e da Tabela 1 mostradas a seguir, as contagens das publicações são relatadas como tendo 2012 como ano de referência, usando as equações empregadas no cálculo de inflação monetária. A inflação $r(y)$ para o ano y é calculada usando $\bar{c}(2012)$, a contagem média de publicações por pessoa em 2012, e a contagem média do ano y , $\bar{c}(y)$, de acordo com a Equação 1.

$$r(Y) = \frac{\bar{c}(2012) - \bar{c}(Y)}{\bar{c}(Y)} + 1 \quad (1)$$

A Figura 2 caracteriza a trajetória média dos pesquisadores com ao menos 30 anos de atividade. Pode-se observar, em conjunto com o mapa de calor na Figura 3, que é difícil caracterizar a trajetória dos 1000 (mil) pesquisadores mais produtivos da área conforme a narrativa canônica de pico-e-decréscimo.

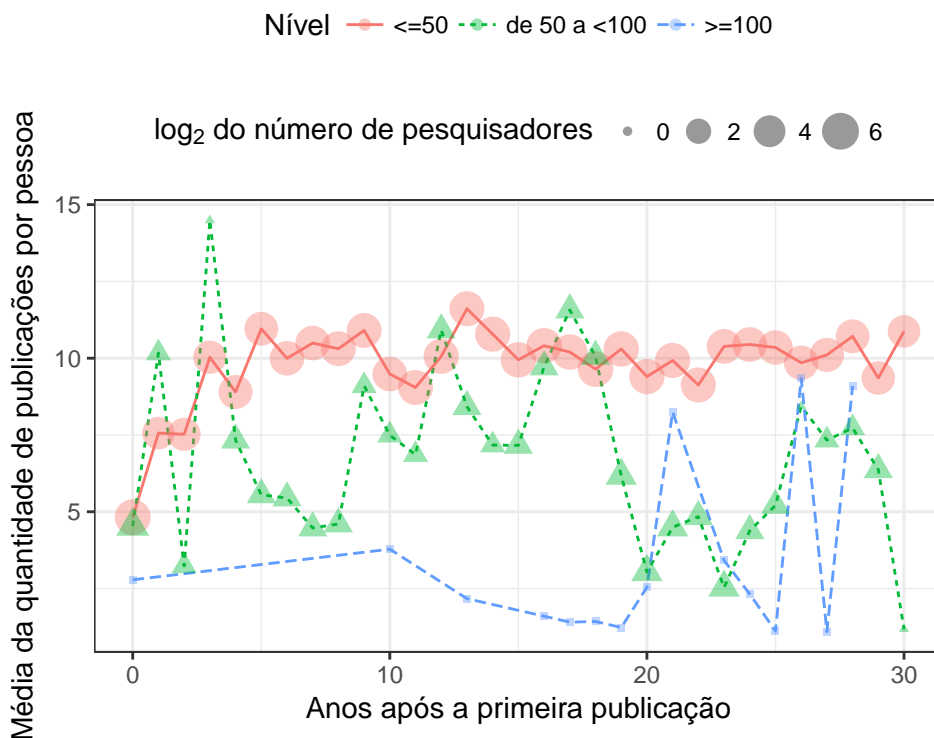


Figura 2. Trajetórias médias de pesquisadores com pelo menos 30 anos de carreira não seguem uma forma de declínio máximo. As trajetórias selecionadas para este número possuem pelo menos uma publicação (de qualquer tipo) antes de 1988.

3. Taxas de publicação e prestígio

Estudos prévios descobriram que pesquisadores de instituições mais prestigiadas tendem a ser mais produtivos [Crane 1965, Reskin 1979, Zuckerman 1970, Long 1978, Allison and Long 1990, Pelz 81].

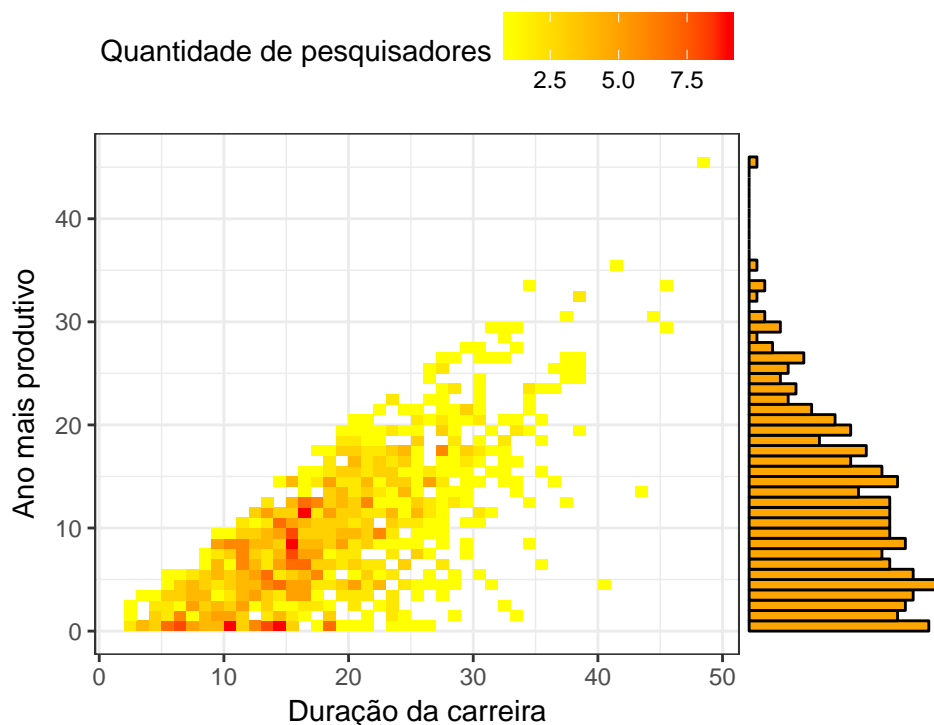


Figura 3. Mapa de calor do ano-pico para os 1000 pesquisadores mais produtivos no campo de IA.

A Figura 4 mostra a mediana das publicações por instituição, as quais foram estratificadas entre públicas e privadas. Pode-se observar que nossos dados corroboram com tais estudos.

A Tabela 1 apresenta as 50 (cinquenta) primeiras instituições ordenadas de acordo com a mediana da quantidade de publicações produzidas. Das quais apenas 15 são privadas (30%). Porém, a instituição com a maior mediana calculada é a Pontifícia Universidade Católica do Rio de Janeiro, com uma mediana de 141.59 publicações.

4. Caracterização de tópicos de interesse

Scholkopf et al. [Schölkopf et al. 2000] desenvolveram um método de classificação *one-class*³. Esse método tem como objetivo estimar uma função f , Equação 2, que reconhece, com probabilidade de erro ν , vetores que pertencem ao espaço dos vetores de treinamento. Essa função usa um kernel $k(x_i, x)$ ponderado por constantes α_i , a serem encontradas por meio de otimização e que definem os vetores-suportes. Esses vetores podem ser interpretados como que definem as principais tendências dos dados de treino. No contexto deste artigo, eles representam os títulos dos artigos que subsumem os temas e assuntos

³Classificação SVM *one-class* é implementada no pacote R *e1071* utilizando o argumento *type* igual a *one-classification* [Meyer et al.].

Tabela 1. Mediana das contagens da publicação de doutores com pelo menos 10 anos de experiência. Contagens são ajustadas pela inflação.

	Instituição	Ranking	Mediana	Tipo	Doutores	Sigla
1	Pontifícia Universidade Católica do RJ	15	141.59	privada	19	PUCRIO
2	Universidade Regional de Blumenau	88	134.85	pública	2	FURB
3	Universidade Presbiteriana Mackenzie	51	128.02	privada	7	UPM
4	Universidade Federal do Rio de Janeiro	3	127.46	pública	32	UFRJ
5	Universidade Santa Cecília	144	117.43	privada	1	UNISANTA
6	Universidade Estadual de Campinas	2	111.27	pública	28	UNICAMP
7	Universidade Federal de Santa Catarina	6	102.03	pública	42	UFSC
8	Pontifícia Universidade Católica do Campinas	78	95.84	privada	4	PUCAMP
9	Universidade Federal do Maranhão	48	92.97	pública	12	UFMA
10	Pontifícia Universidade Católica do Paraná	33	89.87	privada	14	PUCPR
11	Universidade Católica de Pelotas	123	85.28	privada	1	UCPel
12	Instituto Tecnológico de Aeronáutica	59	81.82	pública	12	ITA
13	Universidade Federal do Rio Grande do Sul	4	81.19	pública	33	UFRGS
14	Universidade Federal do Pará	21	80.86	pública	12	UFPA
15	Universidade Federal de Pernambuco	12	79.11	pública	43	UFPE
16	Universidade Estadual do Oeste do Paraná	62	78.07	pública	8	UNIOESTE
17	Universidade do Vale do Itajaí	74	77.76	privada	8	UNIVALI
18	Universidade Católica Dom Bosco	156	72.35	privada	1	UCDB
19	Universidade do Estado do Rio de Janeiro	17	69.27	pública	20	UERJ
20	Universidade do Vale do Rio dos Sinos	41	66.85	privada	10	UNISINOS
21	Universidade de São Paulo	1	66.34	pública	99	USP
22	Universidade Federal de Viçosa	25	66.32	pública	7	UFV
23	Universidade Federal do Espírito Santo	31	65.5	pública	13	UFES
24	Universidade Estadual do Ceará	58	60.5	pública	4	UECE
25	Universidade Federal Fluminense	11	57.99	pública	24	UFF
26	Universidade Estadual de Londrina	23	57.69	pública	4	UEL
27	Universidade Nove de Julho	63	57.54	privada	5	UNINOVE
28	Universidade de Fortaleza	75	57.48	privada	7	UNIFOR
29	Universidade Federal de Minas Gerais	5	56.16	pública	20	UFMG
30	Universidade Federal de Sergipe	47	55	pública	5	UFS
31	Universidade de Pernambuco	89	54.13	pública	6	UPE
32	Universidade de Caxias do Sul	60	53.08	privada	7	UCS
33	Univ. Est. Paulista Júlio de Mesquita Filho	7	52.22	pública	22	UNESP
34	Universidade Federal do ABC	32	52.12	pública	11	UFABC
35	Universidade Federal de Campina Grande	40	50.8	pública	17	UFCG
36	Univ. Fed. de Ciências da Saúde de Porto Alegre	105	50.49	pública	3	UFCSPA
37	Universidade Federal do Paraná	8	49.7	pública	23	UFPR
38	Universidade de Santa Cruz do Sul	64	49.51	privada	3	UNISC
39	Universidade Federal do Rio Grande do Norte	19	48.91	pública	27	UFRN
40	Universidade Federal do Estado do RJ	69	48.72	pública	8	UNIRIO
41	Universidade Federal de Santa Maria	22	47.63	pública	14	UFSM
42	Universidade Federal de São Carlos	16	46.63	pública	21	UFSCAR
43	Universidade Federal da Paraíba	24	46.03	pública	9	UFPB
44	Fundação Getúlio Vargas	28	45.99	privada	3	FGV
45	Universidade Federal do Ceará	10	45.54	pública	14	UFC
46	Universidade Federal da Fronteira Sul	110	44.86	pública	2	UFFS
47	Universidade Católica de Brasília	54	44.8	privada	6	UCB
48	Universidade Federal de Lavras	36	44.06	pública	4	UFLA
49	Universidade Federal Rural da Amazônia	66	43.61	pública	3	UFRA
50	Centro Fed. de Educação Tecnológica de MG	96	43.37	pública	9	CEFET-MG

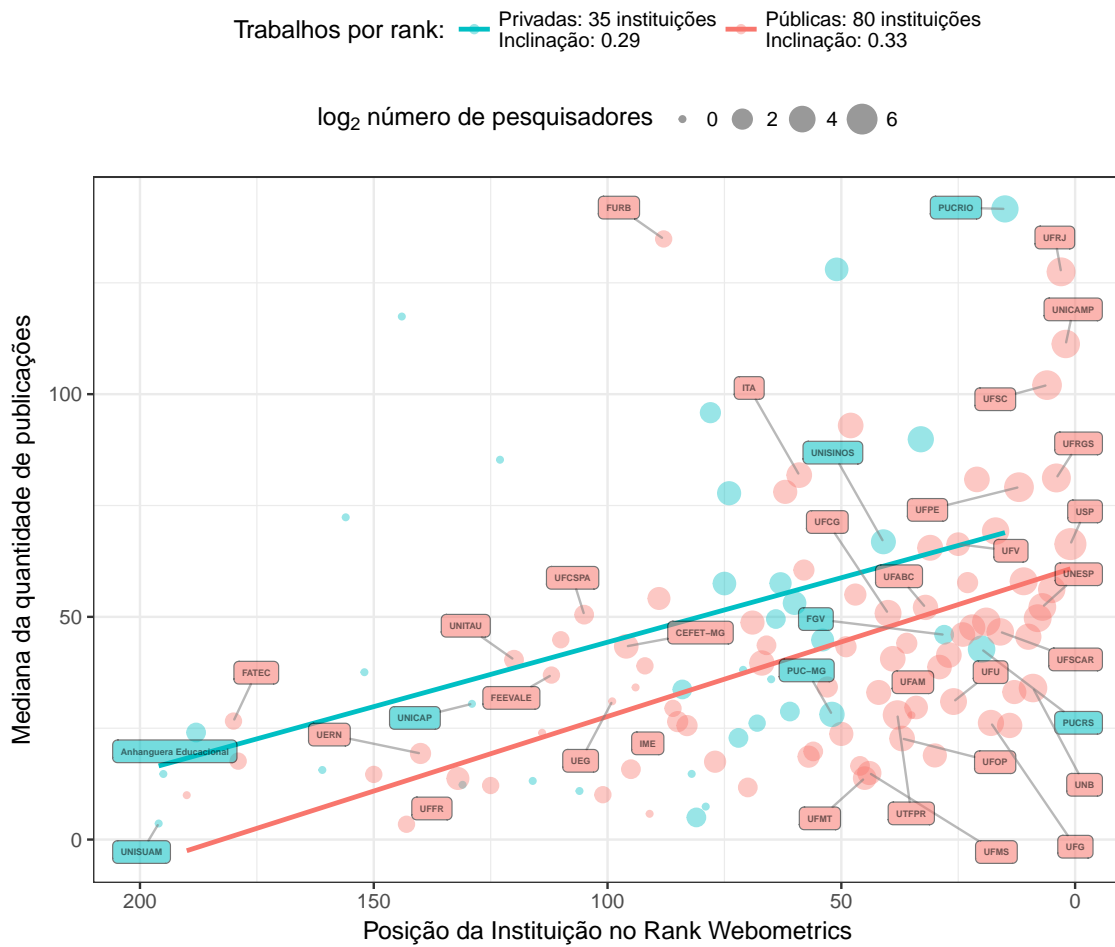


Figura 4. Contagem da mediana das publicações dos doutores correlacionadas com o prestígio das instituições em que trabalham. Os pontos indicam a mediana das publicações ajustadas pela inflação. As linhas são regressões lineares ponderadas pelo número de pesquisadores em cada instituição.

abordados no corpus de publicações dos pesquisadores em IA.

$$f(x) = \text{sign} \left(\sum_i \alpha_i k(x_i, x) - \rho \right) \quad (2)$$

A formulação para o problema de otimização para encontrar as constantes α_i , que definem a importância de cada vetor-suporte, é apresentada nas Equações 3 e 4. Essa formulação segue o paradigma das *Support Vector Machines* e pode ser resolvida por meio de um algoritmo de otimização quadrático. Sendo que nesse caso, o vetor w faz o papel de separador do que é e não é o espaço dos vetores de treinamento. As variáveis ξ_i definem as folgas para se considerar vetores como *outliers* do conjunto de treinamento.

$$\min_{w \in F, \xi \in \mathbb{R}^l, \rho \in \mathbb{R}} \frac{1}{2} \|w\|^2 + \frac{1}{vl} \sum_i \xi_i - \rho \quad (3)$$

$$\text{sujeito a } (w \cdot \Phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0 \quad (4)$$

4.1. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) é um modelo frequentemente aplicado à compreensão de documentos textuais [Blei et al. 2003, Liu and Zhang 2012].

Esse modelo pressupõe que os termos presentes em documentos de um corpus são probabilisticamente explicados por um conjunto de tópicos geradores.

Sendo o LDA útil para modelagem de tópicos, sua abordagem baseia-se em criar uma distribuição de grupos para cada termo de um documento textual e uma distribuição de grupos para cada documento. Baseado nessas distribuições pode-se agrupar documentos de acordo com as probabilidades associadas a cada grupo. Por meio da distribuição dos grupos para os termos dos documentos do corpus (o vocabulário) o LDA também pode ser aplicado para se extrair os termos de maior influência na formação de cada tópico. Como exemplo, um tópico sobre computação será aquele que contém termos relacionados à computação com maior probabilidade de ocorrência. Em contraposição, um tópico que se relacione com qualquer outro assunto distinto conterà termos sobre computação com probabilidade de ocorrência muito baixa ou zero.

O processo de geração de documentos com LDA é realizado em duas etapas, na forma de um algoritmo do tipo *Expectation-Maximization* [Blei 2012]. Esse algoritmo visa estimar os parâmetros da distribuição de probabilidades que define $P(\beta_{1:k}, \theta_{1:D}, Z_{1:D}, W_{1:D})$, tal que os tópicos são representados por $\beta_{1:k}$, onde cada β_k é uma distribuição sobre o vocabulário. As proporções dos tópicos para o d -ésimo documento são θ_d , onde $\theta_{d,k}$ é a proporção de tópicos para o tópico k no documento d . As atribuições dos tópicos para o d -ésimo documento são Z_d , onde $Z_{d,n}$ é a atribuição do tópico para a n -ésima palavra no documento d . Finalmente, as palavras observadas para o documento d são W_d , onde $W_{d,n}$ é a n -ésima palavra no documento d , que é um elemento do vocabulário [Blei 2012].

Cada documento d do corpus apresentará tópicos ($\beta_{1:k}$) com proporções diferentes (θ_d). Cada termo em cada documento é obtido a partir de um dos tópicos ($Z_{d,n}$), que por sua vez é escolhido a partir da distribuição sobre os tópicos de um documento em particular. Esse modelo reflete a intuição de que documentos exibem múltiplos tópicos.

5. Extração de tópicos de interesse

Os seguintes passos foram utilizados com o objetivo de se extrair os termos mais prováveis dos tópicos de pesquisa em Inteligência Artificial.

Primeiramente, foram encontradas 91376 publicações declaradas pelos 2967 currículos de pesquisadores registrados no sistema Lattes. A partir dos dados bibliográficos dessas publicações, efetuou-se a extração do título de cada publicação dos pesquisadores atuantes em IA. O corpus dos títulos foi usado para se obter o vocabulário global das publicações.

Os termos para construção do vocabulário passaram por uma etapa de pré-processamento na qual foram aplicadas a extração de radicais de termos (*stemming*) e a remoção de *stopwords* em Português e Inglês. Nessa etapa foram utilizados

os pacotes R *quanteda*, disponível em <http://quanteda.io>, para construção de um índice com modelo TF-IDF, e *topicmodels*, que permite a extração de tópicos LDA [Hornik and Grün 2011].

Após esse pré-processamento, obteve-se um corpus contendo um vocabulário com 55568 termos utilizados nos títulos das publicações encontradas.

Esse corpus foi segmentado por cada ano $a \in [1965 \dots 2012]$ tal que, para cada ano a , obteve-se os vetores-suportes de um modelo SVM de classe única com parâmetro $\nu = 2^{-15}$, função *kernel* linear e documentos de treino de 1965 até ano atual a . De acordo com a formulação de Scholkopf et al., o valor baixo para ν implica na existência de uma baixa probabilidade de um título produzido até o ano a não ser representável pelos vetores-suportes obtidos.

Os vetores-suportes do modelo SVM do ano a caracterizam o conhecimento acumulado até o momento. Esses vetores foram utilizados para extrair os k tópicos LDA que selecionam os T termos mais prováveis de cada tópico por ano.

Na Tabela 2 estão apresentados os $T = 7$ termos mais prováveis para cada um dos $k = 10$ tópicos por ano. Devido à limitação de espaço, neste artigo estão apresentados apenas uma parte dos anos mais recentes⁴.

6. Conclusões

Neste artigo estudou-se a evolução da produtividade de pesquisadores brasileiros no campo da Inteligência Artificial, bem como a evolução dos tópicos de suas pesquisas.

Foram apresentadas análises quantitativas sobre a produtividade individual dos pesquisadores, analisando-se a influência dos níveis de prestígio das instituições. Observou-se que quanto maior o prestígio das instituições, mais produtivos são os seus pesquisadores, corroborando assim com estudos anteriores [Crane 1965, Reskin 1979, Zuckerman 1970, Long 1978, Allison and Long 1990, Pelz 81].

Por meio do emprego de um método de classificação *one-class* baseado em Support Vector Machines (SVM) proposto por Scholkopf et al. [Schölkopf et al. 2000], buscou-se obter um subconjunto de títulos caracterizadores das pesquisas desenvolvidas nas publicações dos pesquisadores em Inteligência Artificial (91376 publicações declaradas pelos 2967 currículos). Com tais títulos, aplicou-se um modelo de extração de tópicos com o objetivo de avaliar os principais tópicos, e correspondentes termos, que são assuntos de publicações ao longo dos anos.

Nesses tópicos observou-se que nos anos mais recentes aparecem termos de tópicos mais relevantes na área da Inteligência Artificial do que aqueles artigos produzidos desde a década de 1960 e 1970.

Referências

Allison, P. D. and Long, J. S. (1990). Departmental effects on scientific productivity. *American sociological review*, pages 469–478.

⁴Em <http://www.facom.ufu.br/~albertini/tabela2CompletaENIAC2017.pdf> pode-se obter a tabela completa para todos os anos.

Tabela 2. Tópicos LDA de vetores-suportes que caracterizam publicações.

Ano	Tópico 1	Tópico 2	Tópico 3	Tópico 4	Tópico 5	Tópico 6	Tópico 7	Tópico 8	Tópico 9	Tópico 10
2011	learn network data model map classif tecnic	approach test dynam ontolog architectur interact assist	distribut model process integr gerenc inform web	control ambient virtual plan improv design network	analís imagens network evolut process construct knowledg	atrav logic detect previsa fuzzy genet support	red neur rob mov multiobject aplic genet	ensin aprendizag desenvolv ferrament develop framework analysi	projet aplic imag maquin brasil conhec digit	robot context uso techniqu web learn semant
2009	neural control network design dynam evalu robot	virtual web approach distribut servic text multipl	ontolog informat classifi metod semant databas apoi	model inform classif multiag optim evo lutionari paramet	projet ambient gesta red uso apoi desenvolv	evolut aplic genet cluster conhec develop mineraca	agent aprendizag ensin medic avaliaca ambient process	analysi learn simul architectur local optim manag	process imagens agrup digital programa atrav computaca	red model automat neur mode lag artific comput
2007	process logic projet construca computaca integr tecnolog	comput aprendizag virtual ensin ambient game programaca	network neural control imagens medic web studi	red automat neur approach tool artific semant	ontolog desenvolv ambient agent model mecan computacional	rob genet hybrid evalu model optim ensembl	learn environ manag support reason meta heurist local	model conhec fuzzi distribut ferrament classif composit	design imag color cluster approach network orient	cas metod analís multiagent ferrament monitor languag
2005	model process analís select imagens automat	model simul genet autonom otimizaca rob aplic	virtual ferrament intellig apoi orient engin cas	agent ambient conhec process analysi intelligent gesta	control avaliaca uso aprendizag model regr web	red neur artific utilizaca model previsa part	model control neural artific complex model educ inteligenc	network neural artifici complex model educ strategi	comput structur analysi class implement design red	desenvolv adapt cluster aplic approach web ant
2003	intelligent digital tecnic avaliaca data remot land	neural network potenc metod imag model numer	red adapt ontolog ambient utilizaca arquitetur pesquis	logic fuzzy approach program auxili learn knowledg artifici	agent ambient desenvolv distanc educaca virtu seleca	ferrament environ applic develop approach tutor design	dat mining genet analysi orient intellig cas	conhec desenvolv aplic automat gesta informaca evoluca	model comput artific red neur grid plataform	robot web mei dinam multiagent distribut appli
2001	approach imag process signal analís fuzzi model	comput data set red program integr function	agent utilizaca ambient ensin abor dag analís manag	logic fuzzy dat cooper aplic extraca paraconsis tent	neural network artifici virtual model learn forecast	web model cas comport develop raciocini realidad	ensin intelligenc artificial internet educaca via conhec	control apoi simul architectur metod modelag desenvolv	red neur virtual especial ambient aspect atrav	genet design distribu avaliaca objet aplicaco framework
1999	geograf informaca model banc uso process dat	network neural optim model artifici approach identif	case analysi appli expert studi reason assess	fuzzi inter fac approach design model data gerenc	program logic arquitetur control aplic distribu virtual	comput applic develop educ educaca model project	ambient ensin projet desenvolv model virtual conhec	cooper especial aprendizag integr environ generat virtual	red neur languag modelag natural intelligent ferrament	conhec intellig avaliaca autonom robot adapt navig
1997	program resoluca operaco consult banc logic general	neural modelag analís banc localizaca otim entrepot	automat static dynam analís ofshor structur support	projet tecnolog grup aplic celul fabricaca logic	analís metod element finit sol plataform fix	esquem toleranc cultiv soj trig acidez sol	control projet exempl monitor multitaref temp real	semant aco gmc formal approach pfsmfg petri	cooper trabalh caracterizaca enfoco aprendizag apresentaca cas	control oscil dynam integr comput vision hardwar
1995	approach analís compara raca user interfac ergonom studi	design alto qualít una resist high finit	time analís tensa default reason estim path	comput logic continu multiagent en via approach	red neur modelag ferrament desenvolv knowledg cooper	fuzzi neur via support interval optim assist	neural control ensin apoi musical model taxonomi	process metod logic automat aquisica conhec machin	learn knowledg compari son applic languag acquisit program	expert aplic computaca analís chrysony engenh qualidad

- Barata, R. B. and Goldbaum, M. (2003). A profile of researchers in public health with productivity grants from the Brazilian National Research Council (CNPq). *Cadernos de Saúde Pública*, 19(6):1863–1876.
- Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.
- Cole, S. (1979). Age and scientific performance. *American journal of sociology*, 84(4):958–977.
- Crane, D. (1965). Scientists at major and minor universities: A study of productivity and recognition. *American sociological review*, pages 699–714.
- Digiampietri, L. A., Mena-Chalco, J., Silva, G. S., Oliveira, L., Malheiro, A., and Meira, D. (2012). Dinâmica das relações de coautoria nos programas de pós-graduação em computação no Brasil. In *2012 Brazilian Workshop on Social Network Analysis and Mining*.
- Hornik, K. and Grün, B. (2011). topicmodels: An r package for fitting topic models. *Journal of Statistical Software*, 40(13):1–30.
- Liu, B. and Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer.
- Long, J. S. (1978). Productivity and academic position in the scientific career. *American sociological review*, pages 889–908.
- Mena-Chalco, J. P., Digiampietri, L. A., and Oliveira, L. B. (2012). Perfil de produção bibliográfica dos programas brasileiros de pós-graduação em ciência da computação. *Encontro Brasileiro de Bibliometria e Cientometria*, 3.
- Mena-Chalco, J. P., Junior, C., and Marcondes, R. (2009). Scriptlattes: an open-source knowledge extraction system from the lattes platform. *Journal of the Brazilian Computer Society*, 15(4):31–39.
- Mendes, P. H. C., Martelli, D. R. B., Souza, W. P. d., Quirino Filho, S., and Martelli Júnior, H. (2010). Perfil dos pesquisadores bolsistas de produtividade científica em medicina no CNPq, Brasil. *Rev. bras. educ. méd.*, 34(4):535–541.
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. e1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien. 2015. r package version 1.6-7.
- Pelz, D. (81). Andrews, fm 1976. scientists in organizations: Productive climates for research and development. *Ann Arbor, Mi: Institute for Social Research*.
- Reskin, B. F. (1979). Academic sponsorship and scientists' careers. *Sociology of Education*, pages 129–146.
- Scarpelli, A. C., Sardenberg, F., Goursand, D., Paiva, S. M., and Pordeus, I. A. (2008). Academic trajectories of dental researchers receiving CNPq's productivity grants. *Brazilian Dental Journal*, 19(3):252–256.

- Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., and Platt, J. C. (2000). Support vector method for novelty detection. In *Advances in neural information processing systems*, pages 582–588.
- Simonton, D. K. (1997). Creative productivity: A predictive and explanatory model of career trajectories and landmarks. *Psychological Review*, 104(1):66.
- Symonds, M. R., Gemmell, N. J., Braisher, T. L., Gorringer, K. L., and Elgar, M. A. (2006). Gender differences in publication output: towards an unbiased metric of research performance. *PLoS One*, 1(1):e127.
- Tess, B. H., Furuie, S. A. S., Castro, R. C. A. F., Barreto, M. d. C. C., and Nobre, M. R. C. (2009). Assessing the scientific research productivity of a Brazilian healthcare institution: a case study at the heart institute of Sao Paulo, Brazil. *Clinics*, 64:571 – 576.
- Way, S. F., Morgan, A. C., Clauset, A., and Larremore, D. B. (2016). The misleading narrative of the canonical faculty productivity trajectory. *arXiv preprint arXiv:1612.08228*.
- Zuckerman, H. (1970). Stratification in american science. *Sociological Inquiry*, 40(2):235–257.

Métodos de Otimização Hiperparamétrica: uma Análise Comparativa na Abordagem de Problemas de Classificação Binária Utilizando Floresta Aleatória

Wagner J. Alvarenga Jr.^{1,2}, André P. Lemos^{1,2}

¹Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais - UFMG
Av. Antônio Carlos 6627, 31270-901 - Belo Horizonte, MG - Brasil

²Laboratório de Diagnóstico de Falhas, Controle, Otimização e Modelagem - D!FCOM
Universidade Federal de Minas Gerais - UFMG
Belo Horizonte, MG - Brasil

wagnerjaj@ufmg.br, andrepl@cpdee.ufmg.br

Abstract. *In statistical (and machine) learning, hyperparameters are variables that, in somehow, drive the model space or the adjustment procedure of the model. Their optimum estimation is an inherent effort to obtain the best performance for a model. In the literature exists several methods to deal with this task. This paper studies the following methods: Grid Search, Random Search and Bayesian Optimization (using Gaussian Process and Tree-structure Parzen). These four techniques are discussed here, and also compared for hyperparameter tuning of Random Forest models applied in binary classification problems.*

Resumo. *No treinamento de modelos de aprendizado estatístico (e de máquina), hiperparâmetros são variáveis que, de alguma forma, governam o espaço do modelo ou o procedimento de ajuste do mesmo. Sua determinação ótima é uma tarefa intrínseca para extrair o melhor desempenho de um modelo. Existem na literatura diversos métodos para lidar com este desafio. O presente artigo aborda as técnicas: Busca em Grade, Busca Aleatória e Otimização Bayesiana utilizando Processo Gaussiano e com o uso do Estimador de Parzen com Estrutura em Árvore. Estes quatro métodos são apresentados e comparados na tarefa de otimização hiperparamétrica de modelos de Floresta Aleatória, utilizados no tratamento de problemas de classificação de conjuntos binários.*

1. Introdução

Em aprendizado de máquina, usualmente se lança mão de um certo número de hiperparâmetros que se relacionam: ao pré-processamento de dados, à regularização, ao treinamento e à definição estrutural de um modelo, a fim de obter um melhor desempenho deste último. O uso de hiperparâmetros com valores ótimos, possibilita lograr um modelo com melhor desempenho, evitando assim, problemas como: sobreajuste (*overfitting*¹),

¹O termo *overfitting* é empregado a um modelo com baixo erro de treinamento, porém com alto erro de teste.

modelos simplistas com baixa Capacidade², dentre outros. Ou seja, hiperparâmetros são usados para configurar vários aspectos de um algoritmo de aprendizado, possuindo então uma ampla influência sobre o resultado final do modelo.

Tradicionalmente, a otimização de hiperparâmetros tem sido uma tarefa humana, pela sua grande eficiência quando um pequeno número de valores possíveis são aceitáveis [Bergstra et al. 2011]. Contudo, a exploração manual do espaço destas variáveis, na maior parte dos casos, é uma tarefa tediosa e inclinada a resultar em um desempenho insatisfatório do modelo [Swersky K 2014]. Assim, há atualmente, um grande apelo por parte da comunidade científica, por métodos automáticos para realizar a otimização hiperparamétrica, como por exemplo: a otimização bayesiana [Bergstra et al. 2011, Snoek et al. 2012, Thornton et al. 2013, Swersky K 2014], os algoritmos genéticos [Chunhong and Licheng 2004], os métodos de busca aleatória [Bergstra and Bengio 2012] e de busca em grade [Hsu et al. 2010] e a otimização baseada no gradiente [Bengio 2000, Maclaurin et al. 2015].

O objetivo do presente artigo é abordar os seguintes métodos de otimização hiperparamétrica: busca em grade, busca aleatória, otimização bayesiana (utilizando processo gaussiano) e otimização bayesiana (usando estimador de Parzen com estrutura em árvore), discutindo suas vantagens, desvantagens e peculiaridades. Este artigo possui também em seu escopo, a comparação destes 4 métodos na resolução do problema de otimização hiperparamétrica de modelos de Floresta Aleatória, empregados no problema de classificação binária de padrões.

O trabalho está dividido da seguinte forma: na seção 2. é brevemente exposto o modelo de Floresta Aleatória e alguns de seus hiperparâmetros, na seção 3. é apresentada uma revisão bibliográfica dos métodos de otimização hiperparamétrica: busca em grade, busca aleatória, otimização bayesiana (utilizando processo gaussiano) e estimador de Parzen com estrutura em árvore. Posteriormente, na seção 4., são mostrados os experimentos comparativos entre estes quatro métodos. E por fim, são traçadas as considerações finais na seção 5..

2. Floresta Aleatória

Florestas Aleatórias [Breiman 2001] são modelos do tipo *ensemble methods*, que combinam a predição de um conjunto de árvores de decisão, compostas de simples regras do tipo *Se-Então* (em cada um de seus nós terminais), para obter uma única resposta como saída. Na construção de uma Floresta Aleatória, é utilizado a técnica *Bootstrap* (Efron 1979), que amostra os dados de treinamento com reposição, para criar árvores distintas com diferentes valores de corte e atributos em seus nós terminais. O Pseudocódigo 1 mostra um procedimento para construir um modelo de Floresta Aleatória. Para realizar a classificação de uma nova instância, a partir do modelo treinado, pode-se tomar o voto majoritário entre os preditos por cada árvore do modelo, ou tomar a combinação das probabilidades de cada classe, dadas por cada árvore.

Na construção de uma Floresta Aleatória são utilizados hiperparâmetros que controlam o grau de ajuste do modelo aos dados de treinamento e a variabilidade presente

²Para um melhor desempenho, um modelo deve apresentar uma Capacidade adequada à complexidade do problema apresentado.

Pseudocódigo 1 Construção de uma Floresta Aleatória

para $b=1$ até B **faça** ▷ B é o número total de árvores

a) Amostre os dados de treinamento utilizando a técnica *bootstrap*.

b) Cresça uma árvore T_b com o conjunto amostrado, repetindo-se os próximos passos, para cada nó terminal da árvore, até que o último nó, n_{max} , seja alcançado: ▷ n_{max} define a máxima profundidade da árvore

i. selecione aleatoriamente m atributos do total de p existentes.

ii. escolha o atributo com melhor cenário de divisão do nó entre os m .

iii. divida o nó em dois nós filhos.

fim para

nas respostas das árvores de decisão. Um destes hiperparâmetros é o *número de árvores*, presentes no modelo. Em geral, quanto maior é este valor, melhor a acurácia de predição. Porém, o aumento deste número é benéfico até certo ponto, uma vez que a partir de certa quantidade de árvores, a melhoria na resposta combinada cessa, além do fato de um maior número de árvores consumir uma maior quantidade de memória computacional. O hiperparâmetro *profundidade máxima*³ atua no balanceamento entre subajuste e sobreajuste dos dados de treinamento, controlando a quantidade de regras *Se-Então*. A *quantidade de atributos considerados no split* controla o grau de descorrelação entre as árvores do modelo, na medida que seleciona aleatoriamente um percentual dos atributos para serem considerados em cada *split* durante a criação da Floresta Aleatória. O ajuste desta descorrelação possibilita melhorar a redução de variância na resposta combinada de saída do modelo.

3. Revisão Bibliográfica

3.1. Busca em Grade

A estratégia da busca em grade é um método tradicionalmente utilizado na otimização hiperparamétrica de modelos com um número reduzido de hiperparâmetros. Esta técnica apresenta uma forma de resolução simples e direta, sendo de fácil implementação e paralelização.

No método de busca em grade, cada hiperparâmetro é delimitado em torno de um intervalo particular de busca, no qual acredita-se que seja um potencial local para a varredura. Para cada hiperparâmetro, é estabelecido uma resolução de grade que determina a quantidade de pontos à serem considerados em toda a experimentação (aqui também, levando em conta uma quantidade de candidatos que se acredita ser a mais adequada). A execução deste procedimento gera, no espaço hiperparamétrico, uma estrutura em grade, formada pelo conjunto de vetores dos hiperparâmetros ($\lambda_{1:t}$) que são posteriormente experimentados, um a um, por meio de uma função custo (\mathcal{L}). A finalidade de todo este procedimento é achar o melhor ponto de configuração (λ^*), entre todos os vetores λ_i que são testados.

Uma vez que se tenha delimitado um intervalo do espaço hiperparamétrico, e este captura a região onde habita o valor ótimo de configuração, e que, além disso, tenha-se

³Existem outros hiperparâmetros de uma Floresta Aleatória que podem ser também utilizados no ajuste do modelo, como: *número mínimo de pontos presentes nas folhas (nodos terminais)*, *número mínimo de pontos necessários para a ocorrência de um split (divisão de um nó)*, dentre outros.

escolhido uma resolução fina para cada hiperparâmetro, o método de busca em grade possui grande chance de atingir o valor ótimo, ou muito próximo do ótimo. O problema da utilização do método de busca em grade reside no crescimento exponencial de treinamentos necessários, a medida que o número de hiperparâmetros a serem estimados aumenta.

3.2. Busca Aleatória

Algoritmos de busca aleatória compreendem vários métodos como: Busca Aleatória Pura, Busca Aleatória Adaptativa, *Simulated Annealing*, dentre outros. Em sua forma mais simples (busca aleatória pura), a resolução do problema de otimização hiperparamétrica se dá por meio da amostragem aleatória de valores para cada hiperparâmetro (λ_i) do modelo, utilizando uma distribuição uniforme e um subespaço \mathbb{R}^λ . O método consiste então em uma amostragem seguida pelo treinamento e avaliação do modelo utilizando o vetor com os hiperparâmetros amostrado. Estas duas tarefas acontecem em ciclo, por diversas vezes, até que uma condição de parada ocorra (por ex.: número máximo de treinamentos ou tempo de pré-programado).

Nos experimentos para grandes dimensões em [Bergstra and Bengio 2012], foi observado que o método de busca aleatória é mais eficiente que a busca em grade, ao mesmo tempo que mantém as características de simplicidade, facilidade de implementação e paralelismo. A explicação para esta superioridade se deve ao fato de pontos uniformemente amostrados de modo aleatório, em cada dimensão, podem cobrir densamente cada subespaço de baixa dimensão, como mostrado na Figura 1. O que aumenta as chances de encontrar uma configuração de melhor desempenho.



Figura 1. Exemplo de distribuição típica dos hiperparâmetros quando se utiliza os métodos de busca em grade e busca aleatória. Imagem inspirada em [Bergstra and Bengio 2012].

3.3. Otimização Bayesiana

Para tratar da otimização bayesiana é necessário introduzir primeiramente, alguns conceitos de Estatística Bayesiana. Uma regra muito importante neste contexto é o teorema de Bayes, dado por,

$$p(A|B) = \frac{p(B|A)}{p(B)}p(A) \iff \text{posteriori} = \frac{\text{verossimilhança}}{\text{evidência}}\text{priori}, \quad (1)$$

onde A e B são eventos, sendo $p(B) \neq 0$.

A regra de Bayes descreve uma distribuição *a posteriori* $p(A|B)$, que é a probabilidade condicional de se observar o evento A dado que B seja verdadeiro, a partir do conhecimento da evidência $p(B)$, da probabilidade *a priori* $p(A)$, e do modelo de verossimilhança $p(B|A)$.

Na resolução do problema de otimização hiperparamétrica, pode-se escrever o conjunto de pares formados pelos: hiperparâmetros experimentados em um modelo $(\lambda_{1:t})$ e a avaliação deste último por meio de uma função custo qualquer $(f(\lambda_{1:t}))$, como $\mathcal{D}_{1:t} = (\lambda_{1:t}, f(\lambda_{1:t}))$. Assim, a probabilidade *a posteriori* da função f desconhecida (tomada como um evento) é expressa como,

$$p(f|\mathcal{D}) \propto p(\mathcal{D}|f)p(f), \quad (2)$$

onde $p(f)$ representa a crença sobre o espaço de possibilidades para a função desconhecida, $p(\mathcal{D}|f)$ é a função de semelhança e $p(f|\mathcal{D})$ é a distribuição *a posteriori*.

Para a busca hiperparamétrica, existe na literatura duas abordagens da otimização bayesiana que se destacam em maior grau. Em uma delas assume-se que a função f desconhecida seja amostrada a partir de um Processo Gaussiano (*Gaussian Process* - GP) [Snoek et al. 2012]. Neste caso, o método modela diretamente a distribuição *a posteriori* $p(f|\mathcal{D})$. Uma outra forma é o Estimador de Parzen com Estrutura em Árvore (*Tree-structured Parzen Estimator* - TPE), proposto em [Bergstra et al. 2011], no qual é feito o modelamento de $p(\mathcal{D}|f)$ e $p(f)$ separadamente e de formas distintas. Estas duas abordagens são tratadas a seguir.

3.3.1. Processo Gaussiano

No método de otimização bayesiana utilizando o Processo Gaussiano, um modelo probabilístico é construído utilizando uma distribuição *a priori* sobre a função desconhecida. No problema de otimização hiperparamétrica, este modelo gerado é uma função *surrogate*⁴ que representa o desempenho do modelo de aprendizado. Ao longo da resolução do problema, novos pares de observações $\mathcal{D}_{1:t} = (\lambda_{1:t}, f(\lambda_{1:t}))$ são acumulados, após cada treinamento do modelo de aprendizado. À medida que dados são gerados, a distribuição *a priori* $p(f)$ é combinada com a função de semelhança $p(\mathcal{D}_{1:t}|f)$, para produzir a distribuição *a posteriori* $p(f|\mathcal{D}_{1:t})$. Esta última captura então as crenças atualizadas sobre a função desconhecida, para que alguma técnica de otimização maximize a função de aquisição⁵, de modo a obter o próximo ponto λ a ser avaliado.

O modelo gerado pelo método é dado por,

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')), \quad (3)$$

em que $m : \mathcal{X} \mapsto \mathbf{R}$ é uma função média, e $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbf{R}$ é uma função de covariância,

⁴Um modelo *surrogate* é um modelo construído para obter alguma medida de interesse, que de uma outra forma seria impossível por não se dispor do modelo verdadeiro ou inviável por requerer um infactível tempo de avaliação.

⁵Avalia a perda esperada associada a um ponto específico da função f .

também conhecida como *kernel*. Um exemplo de definição para estas duas funções é,

$$m(x) = \mathbf{m}, \quad (4)$$

$$k_{SE}(x, x') = \sigma_0^2 \exp\left(-\frac{(x - x')^2}{2\sigma_l^2}\right), \quad (5)$$

em que k_{SE} é o *kernel* Exponencial Quadrático (*Squared Exponential* - SE), x representa os hiperparâmetros λ que se busca otimizar, e \mathbf{m} , σ_0^2 , σ_l^2 são hiperparâmetros do modelo do Processo Gaussiano.

A função de covariância codifica o que se assume sobre a função desconhecida. Sua adequada escolha é crucial para uma captura de maior fidedignidade da função desconhecida através do modelo *surrogate* gerado. Os desafios presentes nesta questão estão relacionados à variedade de tipos de funções de covariância existentes e a presença de hiperparâmetros nestas últimas, responsáveis por suas generalizações.

Expressando o conjunto de dados coletados como $\mathcal{D}_{1:t} = (x_{1:t}, \mathbf{f}_{1:t})$, em que $\mathbf{f}_{1:t} = f(x_{1:t})$, e representando por f_{t+1} o valor da função para o próximo ponto x_{t+1} , então, pelas propriedades do Processo Gaussiano, tem-se que $\mathbf{f}_{1:t}$ e f_{t+1} são distribuições Gaussianas conjuntas,

$$\begin{bmatrix} \mathbf{f}_{1:t} \\ f_{t+1} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & k(x_{t+1}, x_{t+1}) \end{bmatrix}\right), \quad (6)$$

onde,

$$\mathbf{K} = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_t) \\ \vdots & \ddots & \vdots \\ k(x_t, x_1) & \dots & k(x_t, x_t) \end{bmatrix}, \quad (7)$$

é a matriz de covariância, em que cada elemento ij representa a covariância entre os elementos i e j de um vetor x .

$$\mathbf{k} = [k(x_{t+1}, x_1) \quad k(x_{t+1}, x_2) \quad \dots \quad k(x_{t+1}, x_t)], \quad (8)$$

e \mathbf{k} é um vetor com a covariância entre x_{t+1} e cada elemento do vetor x .

Então, a distribuição *a posteriori* $p(f|\mathcal{D})$ é dada pela seguinte distribuição Normal,

$$p(f_{t+1}|\mathcal{D}_{1:t}, x_{t+1}) = \mathcal{N}(\mu_t(x_{t+1}), \sigma_t^2(x_{t+1})), \quad (9)$$

onde,

$$\mu_t(x_{t+1}) = \mathbf{m} + \mathbf{k}^T \mathbf{K}^{-1} \mathbf{f}_{1:t}, \quad \text{média a posteriori}, \quad (10)$$

$$\sigma_t^2(x_{t+1}) = k(x_{t+1}, x_{t+1}) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}, \quad \text{incerteza a posteriori}. \quad (11)$$

Um exemplo de distribuição *a posteriori*, para uma função de dimensão 1, pode ser visto na Figura 2. De posse da distribuição *a posteriori* atualizada, o próximo passo consiste na determinação do próximo ponto x_{t+1} . Isto é feito por meio do uso de uma Função de Aquisição que guia a busca pelo ponto ótimo, balanceando o Aproveitamento da escolha, que visa valores mais altos no modelo (representando melhores desempenhos), e a Exploração, que alveja locais com maior incerteza (variância) da distribuição *a posteriori* [Brochu et al. 2010]. As funções de aquisição tradicionalmente preferidas na otimização bayesiana são: Probabilidade de Melhoria, Esperança de Melhoria e Limite de Confiança Superior.

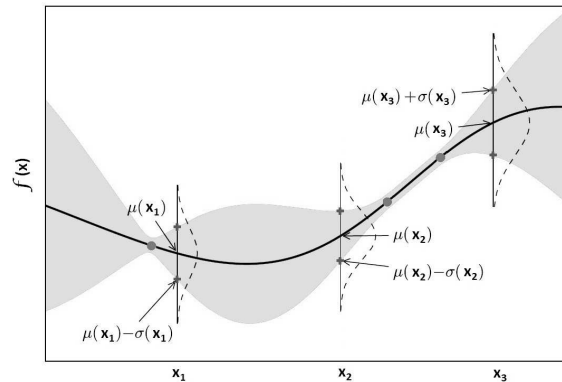


Figura 2. Exemplo de distribuição *a posteriori* de um processo Gaussiano de dimensão 1, com 3 pares (x, f) . A linha escura mostra a média *a posteriori*, e a área sombreada a variância. Imagem reproduzida de [Brochu et al. 2010].

3.3.2. Estimador de Parzen com Estrutura em Árvore

No método TPE, o modelamento de $p(f)$ é realizado individualmente para cada um dos hiperparâmetros do modelo. Nos casos em que estes são contínuos ou discretos, pode-se utilizar uma função de probabilidade, como por exemplo: uniforme, normal, log-normal e log-normal, sendo o valor arredondado para o inteiro mais próximo, para o caso discreto. Para hiperparâmetros categóricos, uma opção é o uso de um vetor que contenha as probabilidades de cada categoria. A Figura 3 mostra estas distribuições. O método

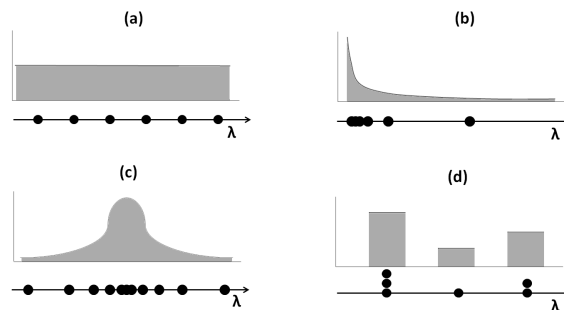


Figura 3. Típicos modelos *a priori* utilizados no método TPE (a) uniforme, (b) log-normal, (c) normal, (d) categórico.

TPE é baseado na técnica de janelas de Parzen, que é um processo não-paramétrico de estimação de densidade, no qual não se assume nenhuma suposição sobre a distribuição da densidade que se estima [Bergstra et al. 2011].

No modelamento de $p(\mathcal{D}|f)$, a técnica TPE utiliza a estimativa de duas densidades, $l(\lambda)$ e $g(\lambda)$, condicionadas da seguinte maneira,

$$p(\mathcal{D}|f) = \begin{cases} l(\lambda), & \text{se } \mathcal{L}(\lambda) < c^*. \\ g(\lambda), & \text{se } \mathcal{L}(\lambda) \geq c^*, \end{cases} \quad (12)$$

onde c^* é uma porção da função custo do modelo (\mathcal{L}), obtida com a melhor configuração entre os λ_i avaliados até então (ou seja, o λ^+). Assim, $c^* = \gamma \mathcal{L}(\lambda^+)$. O valor de γ

usado em [Bergstra et al. 2011] é 0,15. A densidade $l(\lambda)$ é calculada tomando apenas as observações λ_i cuja função custo (\mathcal{L}) ficou abaixo de c^* , enquanto os demais pontos λ_i são usados no cálculo da densidade $g(\lambda)$. Assim, a densidade $l(\lambda)$ diz respeito a pontos λ_i que resultaram em uma boa configuração, ao passo que a densidade $g(\lambda)$ representa as configurações de menor qualidade.

O termo *tree*, presente no nome do método TPE, deve-se ao fato da estrutura hierárquica do espaço hiperparamétrico do modelo otimizado. Este arranjo em árvore conta com hiperparâmetros contínuos, discretos e categóricos, sendo a estimação das densidades $l(\lambda)$ e $g(\lambda)$ realizadas para cada uma destas variáveis, utilizando o método de janela de Parzen. A Figura 4 ilustra este cenário. Em [Bergstra et al. 2011], os autores

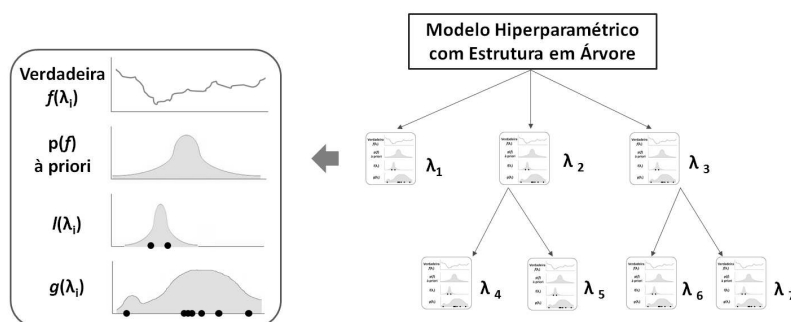


Figura 4. Esquema de um espaço hiperparamétrico com estrutura em árvore. O quadro em destaque mostra a verdadeira $f(\lambda_i)$ (desconhecida), a distribuição a priori $p(\lambda_i)$ escolhida, e as densidades $l(\lambda)$ e $g(\lambda)$ estimadas a partir de alguns pontos (λ_i) já avaliados.

mostraram que a Esperança de Melhoria ($EI_{c^*}(\lambda)$), tomada com base no valor de c^* , é proporcional à expressão,

$$EI_{c^*}(\lambda) \propto \left(\gamma + \frac{g(\lambda)}{l(\lambda)}(1 - \gamma) \right)^{-1}, \quad (13)$$

na qual percebe-se que a minimização do erro de ajuste do modelo depende de pontos λ_i com alta probabilidade em $l(\lambda)$ e baixa probabilidade em $g(\lambda)$. Para realizar tal busca, pode-se utilizar uma simples geração de pontos aleatórios λ_i (no correto espaço hiperparamétrico), de modo a determinar o λ com menor valor para $\frac{g(\lambda)}{l(\lambda)}$. Este cálculo é realizado para cada nó da árvore (Figura 4). Para avaliar uma configuração dada por um λ_i , realiza-se o cálculo partindo do topo da árvore em direção às suas folhas, passando apenas pelos nós referentes ao λ_i . Então o resultado final é obtido com a combinação das probabilidades de cada nó presente no caminho percorrido (entre topo e folhas).

4. Experimentos

Os testes realizados no presente artigo compararam as técnicas: busca em grade, busca aleatória, otimização bayesiana (GP) e (TPE), na otimização de 4 hiperparâmetros presentes em modelos de Floresta Aleatória. Estes modelos foram empregados na resolução de problemas de classificação, utilizando 10 conjuntos de dados binários. Para realizar a comparação dos métodos, utilizou-se duas medidas distintas, sendo: o desempenho

de classificação dos modelos, por meio da métrica AUC, e o tempo, em segundos, despendido durante o treinamento dos modelos. A descrição dos conjuntos de dados, com número de instâncias, atributos e balanceamento (valores nominal e percentual), se encontra na Tabela 1.

Tabela 1. Conjunto de dados binários utilizados nos experimentos.

<i>Conjunto de Dados</i>	<i>Instâncias</i>	<i>Atributos</i>	<i>Balanceamento²</i>
Breast Cancer	569	30	212 (37%) - 357 (63%)
Cardiotocography ¹	2.126 (-176)	23	295 (15%) - 1655 (85%)
Diabetes	1.151	20	540 (47%) - 611 (53%)
German Credit	1.000	20	300 (30%) - 700 (70%)
Ionosphere	351	34	126 (36%) - 225 (64%)
Liver	345	6	145 (52%) - 200 (58%)
Pima Indians	768	8	268 (35%) - 500 (65%)
Qsar	1.055	41	356 (34%) - 699 (66%)
Sonar	208	60	97 (47%) - 111 (53%)
Spambase	4.601	57	1.813 (39%) - 2.788 (61%)

¹ Utilizou-se apenas as duas classes com maior número de instâncias.

² Valores: nominal (percentual) de cada classe.

Cada conjunto de dados foi tratado separadamente por todos os 4 métodos de otimização hiperparamétrica. Para cada método, o treinamento do modelo foi simulado 30 vezes, por existir variabilidade no desempenho de classificação de um mesmo modelo, devido: 1- à divisão aleatória dos conjuntos de dados, em conjunto de treinamento (80%) e teste (20%), de modo estratificado, 2- ao processo de validação cruzada com embaralhamento aleatório dos dados (utilizando 5 dobras), 3- ao procedimento de criação do modelo de Floresta Aleatória e 4- à própria variabilidade existente nos métodos de busca aleatória e otimização bayesiana (GP e TPE). Para estes três últimos métodos, foi configurado a procura por 7 modelos⁶ dentro de cada uma das 30 repetições. E para o método de busca em grade, foi definido um conjunto de hiperparâmetros compostos de 3 pontos (limite inferior, ponto intermediário e limite superior) para cada hiperparâmetro. Como o número de hiperparâmetros considerados nos experimentos é igual a 4, este arranjo se traduz em 81 combinações diferentes para o método de busca em grade. Limites inferior e superior, neste caso, são os mesmos valores limiares definidos para a busca hiperparamétrica nos demais três métodos. A Tabela 2 mostra os hiperparâmetros considerados nos testes, bem como os seus valores.

Tabela 2. Hiperparâmetros configurados nos modelos de Floresta Aleatória e respectivos valores/intervalos de busca adotados.

<i>Hiperparâmetros</i>	<i>Limite Inferior</i>	<i>Intermediário¹</i>	<i>Limite Superior</i>
Número de Árvores	10	1.000	2.500
Profundidade máxima	3	11	20
Atributos no <i>split</i> (%)	10	50	90
Mínimo de pontos para <i>split</i>	2	8	15

¹ Fez-se uso do valor Intermediário apenas no método de busca em grade.

Os testes realizados com o método de busca aleatória utilizaram a função de densidade Uniforme para amostrar aleatoriamente os valores de cada hiperparâmetro, respei-

⁶ Em [Bergstra and Bengio 2012], utilizou-se o treinamento de 8 modelos em cada repetição.

tando o limite informado na Tabela 2. O método TPE foi configurado para usar a função de densidade Uniforme para amostrar os valores referentes à $p(f)$, e para γ foi utilizado o valor padrão de 0,15. Para o método GP foi feita a escolha pela função de aquisição Limite de Confiança Superior, com $\kappa = 5$ (κ é um hiperparâmetro que controla o balanceamento entre Exploração e Aproveitamento). E para *kernel*, foi escolhida a função Matérn, com $\nu = 2,5$, como em [Eggensperger et al. 2013].

Dentro de cada uma das 30 repetições, o modelo de melhor desempenho, dentre os 7 (ou 81, no caso do busca em grade) que foram treinados com validação cruzada, foi então separado dos demais para ser testado com o conjunto de dados de teste. O resultado com estas 30 predições é mostrado por meio da média e desvio padrão da medida AUC na Tabela 3. A Tabela 4 apresenta o tempo total gasto, em segundos, do treinamento dos 7 modelos (ou 81 para a técnica de busca em grade), também por meio da média e desvio padrão, em cada uma das 30 repetições.

Em todos os testes realizados, não houve pré-processamento de dados. Como critério de partição dos nós na construção das Florestas Aleatórias, foi escolhido o índice de Gini, $G = \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk})$, que mede a variância total entre as K classes. Os ex-

Tabela 3. Resultado da Predição utilizando dados de teste e métrica AUC. Valores apresentados são: Média±Desvio Padrão.

Conjunto de Dados	Grade	Métodos		
		Aleatória	TPE	GP
Breast Cancer	0,952±0,018	0,948±0,021	0,957±0,018	0,955±0,015
Cardiotocography	0,874±0,029	0,880±0,027	0,868±0,026	0,873±0,032
Diabetes	0,743±0,031	0,750±0,032	0,748±0,026	0,746±0,023
German Credit	0,798±0,032	0,807±0,031	0,785±0,028	0,801±0,027
Ionosphere	0,905±0,039	0,916±0,032	0,904±0,043	0,912±0,039
Liver	0,772±0,052	0,775±0,048	0,756±0,044	0,754±0,046
Pima Indians	0,826±0,030	0,829±0,027	0,835±0,021	0,822±0,028
Qsar	0,835±0,030	0,836±0,025	0,841±0,027	0,833±0,027
Sonar	0,841±0,053	0,832±0,059	0,818±0,066	0,825±0,046
Spambase	0,948±0,007	0,935±0,013	0,937±0,008	0,943±0,010

Tabela 4. Tempo de treinamento, em segundos, dos 7 modelos treinados para métodos busca aleatória, TPE e GP, e 81 para busca em grade. Valores apresentados são: Média±Desvio Padrão.

Conjunto de Dados	Grade	Métodos		
		Aleatória	TPE	GP
Breast Cancer	979,1±15,0	95,7±25,0	193,9±39,1	256,4±23,9
Cardiotocography	1.101,5±20,1	118,5±26,9	250,7±61,0	290,7±24,9
Diabetes	1.102,3±51,7	106,0±19,3	259,0±54,4	269,6±25,6
German Credit	939,3±47,8	98,9±19,2	196,4±53,1	267,6±28,2
Ionosphere	933,8±12,4	93,5±17,6	186,1±46,7	251,4±29,8
Liver	825,4±13,9	84,9±21,1	179,2±31,9	282,7±40,5
Pima Indians	983,0±35,0	83,1±16,1	194,1±36,4	272,0±29,6
Qsar	1.051,7±11,2	106,8±25,4	241,1±63,1	261,7±28,6
Sonar	917,7±15,7	78,7±16,8	201,0±44,0	249,6±48,2
Spambase	2.213,9±30,7	213,8±61,4	558,3±161,1	330,6±49,8

perimentos foram implementados⁷ utilizando a linguagem de programação Python, com o uso das bibliotecas: Sklearn⁸ (busca em grade e aleatória), bayes_opt⁹ (otimização bayesiana - GP) e hyperopt¹⁰ (TPE). Os experimentos foram simulados em um computador Intel Core i7-2670QM 2.20GHz 4 cores, com 16 GB de memória. Todos os conjuntos de dados foram obtidos do repositório UCI - Machine Learning¹¹, com exceção do conjunto Breast Cancer o qual optou-se por usa-lo diretamente da biblioteca Sklearn.

5. Considerações Finais

Nos resultados mostrados na Tabela 4, a grande discrepância dos tempos médios do método de busca em grade se justifica pelo fato deste ter treinado um maior número de modelos em cada repetição. A métrica tempo está diretamente relacionada ao número de treinamentos, que no caso do método de busca em grade obedece a relação de p^h , onde p é o número de pontos em cada hiperparâmetro e h é o número de hiperparâmetros considerados. Ao se comparar as médias de busca aleatória com as técnicas TPE e GP, observa-se diferenças da ordem de 2 vezes mais demorado para TPE, e 3 vezes mais lento para GP. Estes resultados já eram parcialmente esperado, uma vez que na busca aleatória os hiperparâmetros são amostrados aleatoriamente, no método TPE há a amostragem aleatória dos valores, seguida pela geração e posterior combinação das funções de densidade $l(\lambda)$ e $g(\lambda)$. Já o método GP, precisa otimizar a função de aquisição¹².

Ao se analisar os resultados da métrica AUC (Tabela 3) observa-se um cenário de relativa paridade entre os métodos, com diferenças entre as médias menor do que 0,024 AUC, e desvios padrões muito semelhantes entre si. O desempenho do método Grade provavelmente foi inferior ao método Aleatório, devido a resolução da grade (apenas 3 valores para cada hiperparâmetro). A escolha por apenas 3 valores para cada hiperparâmetro foi optada para não aumentar por demais o custo computacional dos experimentos. Um detalhe importante quanto aos métodos TPE e GP, é que ambos possuem seus próprios hiperparâmetros que necessitam configurações (função de distribuição e valor γ para TPE, e *kernel*, função de aquisição e método de otimização desta última, para o método GP). Uma outra questão, ainda sobre estes dois métodos, é que o ato de se treinar uma maior quantidade de modelos, dentro de uma mesma repetição, tem maior efetividade nestas duas técnicas quando comparadas a busca aleatória. Isto porque o simples fato destes dois métodos utilizarem os pontos já experimentados para auxiliar na procura por melhores candidatos, aumenta suas chances de obterem resultados mais acurados para o classificador. Ou seja, configurações diferentes das que foram utilizadas aqui, para os métodos TPE e GP, ou então o uso de um maior número de modelos treinados dentro de cada repetição, pode vir a se traduzir em melhores configurações do classificador quando este é otimizado com TPE ou GP, quando comparado com a técnica de busca aleatória.

Este artigo abordou os métodos de otimização hiperparamétrica: busca em grade, busca aleatória e otimização bayesiana utilizando processo gaussiano e técnica de estima-

⁷Código disponível em: <https://github.com/Wagner-Alvarenga/Hyperparameter-Optimization/> a partir de 02/10/17.

⁸<http://scikit-learn.org>

⁹<https://github.com/fmfn/BayesianOptimization>

¹⁰<https://github.com/hyperopt/hyperopt>

¹¹<http://archive.ics.uci.edu/ml/>

¹²A biblioteca bayes_opt utilizado o algoritmo "L-BFGS-B" como método de otimização.

dor de Parzen com estrutura em árvore. Foram apresentados e discutidos os resultados dos experimentos comparativos utilizando o problema de otimização hiperparamétrica de modelos de Floresta Aleatória, na resolução da tarefa de classificação de conjuntos binários.

6. Agradecimentos

O presente trabalho foi realizado com o apoio financeiro da CAPES - Brasil.

Referências

- Bengio, Y. (2000). Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900.
- Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(1):281–305.
- Bergstra, J. S., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyperparameter optimization. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.
- Chunhong, Z. and Licheng, J. (2004). Automatic parameters selection for svm based on ga. In *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, volume 2, pages 1869–1872. IEEE.
- Eggenberger, K., Feurer, M., Hutter, F., Bergstra, J., Snoek, J., Hoos, H., and Leyton-Brown, K. (2013). Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In *NIPS workshop on Bayesian Optimization in Theory and Practice*, volume 10.
- Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2010). A practical guide to support vector classification.
- Maclaurin, D., Duvenaud, D., and Adams, R. P. (2015). Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine Learning*.
- Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc.
- Swersky K, Snoek J, P. A. R. (2014). Freeze-thaw bayesian optimization. *ArXiv e-prints*.
- Thornton, C., Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2013). Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 847–855, New York, NY, USA. ACM.

A Study of Neural Networks Models applied to Natural Language Inference

Victor G. Noronha¹, João C. P. da Silva¹

¹Computer Science Department – Federal University of Rio de Janeiro
Rio de Janeiro, RJ – Brazil

Abstract. *Natural Language Inference is a task that given two sentences, a premise \mathcal{P} and hypothesis \mathcal{H} , try to establish entailment, contradiction or neutral relationships between them. The Stanford Natural Language Inference (SNLI) is a corpus that due to its size allows that neural-network based models perform competitively on SNLI benchmark with respect to other approaches to NLI. In this work, we revisit the original neural network-based models described on the original paper and propose some variants to those models, in order to analyze the impact of such modifications on their performance. The proposed modifications were driven by the plenty of results achieved in other works, not constrained to natural language problems, that use the so-called deep learning techniques, hoping this will bring a better understanding of the elements that make up such systems.*

1. Introduction

Natural Language Inference (NLI) is a task that given two sentences, a *premise* \mathcal{P} and *hypothesis* \mathcal{H} , try to establish one of the following relationships between them: (i) *entailment*, when \mathcal{H} is true whenever \mathcal{P} is true; (ii) *contradiction*, when \mathcal{H} is false if \mathcal{P} is true and (iii) *neutral*, when neither entailment nor contradiction can be determined. For example, the sentences \mathcal{P} : *A man inspects the Uniform of a figure in some East Asian Country* and \mathcal{H} : *The man is sleeping* contradict each other, while \mathcal{H} : *A soccer game with multiple males playing* is entailed from \mathcal{P} : *Some men are playing a sport*. [Bowman et al. 2015]

A NLI system can be used in many applications in Natural Language Processing such as [Maccartney 2009]: (i) ensure that a summarized document does not contain any sentences that could be inferred from the rest of the summary, and thus eliminating redundancies; (ii) determine a semantic equivalence between two translations, the reference translation and the candidate one, even if the words that compose the translations are quite different. If the candidate translation and the reference one entail each other, then the candidate is probably a good translation.

The Stanford Natural Language Inference (SNLI) [Bowman et al. 2015] corpus provides 570,152 labeled sentence pairs and, according to [Bowman et al. 2015], is two orders of magnitude larger than all other resources of its type, allowing that a neural-network based model performs competitively on SNLI benchmark with respect to other approaches to NLI, such as rule-based systems, simple linear classifiers.

In this work, we revisit the original neural network-based models described on [Bowman et al. 2015], that were used to produce distributed representations of sentence meaning. We propose some variants to the baseline models presented in

[Bowman et al. 2015] in order to analyze the impact of such modifications on the performance of those models. The proposed modifications were driven by the plenty of results achieved in other works, not constrained to natural language problems but also on image processing tasks, that use the so-called deep learning techniques, hoping this will bring a better understanding of the elements that make up such systems.

2. Neural Network Algorithms and Techniques

2.1. LSTM with Variational Dropout

Recurrent Neural Networks (RNNs) [Elman 1990] are used for modelling time series, using their recurrent structure to retain (past) information which allows one to discover temporal interrelationships between distant events in a temporal sequence of events. Unfortunately, this kind of model is hard to train because it overfits quickly, due to its *vanishing gradient* and *exploding gradient* problems, described in [Bengio et al. 1994].

In order to address this problem, more complex structures were proposed, like the LSTM (*Long Short-Term Memory*) [Hochreiter and Schmidhuber 1997], that is explicitly designed to avoid gradient problems. Another possible way to address these problems is to apply *dropout* [Srivastava et al. 2014], a technique that randomly removes neurons from a specific layer, along with their connections, in order to train potentially different networks on each training step, and thus reducing overfitting. But the technique has never been applied successfully to RNNs [Gal and Ghahramani 2016]. A variant of dropout to RNNs is proposed on [Gal and Ghahramani 2016] and applies dropout on the input and recurrent connections of the RNNs, in particular on LSTM model. The standard LSTM equations are (*sigm* is the Logistic function and \circ is the element-wise product):

$$\begin{aligned}
 \mathbf{i} &= \text{sigm}(\mathbf{h}_{t-1}\mathbf{U}_i + \mathbf{x}_t\mathbf{W}_i) \\
 \mathbf{f} &= \text{sigm}(\mathbf{h}_{t-1}\mathbf{U}_f + \mathbf{x}_t\mathbf{W}_f) \\
 \mathbf{o} &= \text{sigm}(\mathbf{h}_{t-1}\mathbf{U}_o + \mathbf{x}_t\mathbf{W}_o) \\
 \mathbf{g} &= \text{tanh}(\mathbf{h}_{t-1}\mathbf{U}_g + \mathbf{x}_t\mathbf{W}_g) \\
 c_t &= \mathbf{f} \circ c_{t-1} + \mathbf{i} \circ \mathbf{g} \\
 h_t &= \mathbf{o} \circ \text{tanh}(c_t)
 \end{aligned} \tag{1}$$

where \mathbf{i} , \mathbf{f} , \mathbf{o} , \mathbf{g} , c_t , h_t , are the input gate, forget gate, output gate, input modulation gate, cell state and hidden state, respectively. \mathbf{x}_t and \mathbf{h}_{t-1} are the input and the previous hidden state vectors, respectively. With $\omega = \{\mathbf{W}_i, \mathbf{U}_i, \mathbf{W}_f, \mathbf{U}_f, \mathbf{W}_o, \mathbf{U}_o, \mathbf{W}_g, \mathbf{U}_g\}$ input and recurrent weight matrices for each gate, respectively, we can rewrite the gates equations on a vectorized form, explicitly including the dropout on the input and on the recurrent connections, as follows:

$$\begin{pmatrix} \mathbf{i} \\ \mathbf{f} \\ \mathbf{o} \\ \mathbf{g} \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \text{tanh} \end{pmatrix} \left(\begin{pmatrix} \mathbf{x}_t \circ \mathbf{z}_x \\ \mathbf{h}_{t-1} \circ \mathbf{z}_h \end{pmatrix} \cdot \omega \right)$$

(3)

where \mathbf{z}_x and \mathbf{z}_h are the random dropout masks applied to the input and recurrent connections.

2.2. Batch Normalization

Batch Normalization [Ioffe and Szegedy 2015] is a technique for controlling a phenomenon called *internal covariate shift*, which concerns the change of distribution of the layer's inputs, as the parameters of the previous layers change. The technique standardizes the layer inputs, reducing the effects of the changes on the previous layers. According to [Cooijmans et al. 2016], Batch Normalization takes the problem of optimize the neural network parameters to a better-conditioned situation.

The statistics (mean and variance) are calculated over a training mini-batch of size m , for each training step. The input is normalized and squashed through an affine transformation with γ and β parameters to be learned. The idea behind this linear transformation is that if there is a better distribution than the standardized normal, the normalized input will follow this distribution. The equations for a mini-batch $B = \{x_1, \dots, x_m\}$ are:

$$\begin{aligned}\mu_B &= \frac{1}{m} \sum_{i=1}^m x_i \\ \sigma_B^2 &= \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \\ \hat{x}_i &= \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \\ y_i &= \gamma \hat{x}_i + \beta\end{aligned}\tag{4}$$

where y_i is taken as input for the next layer and ϵ is a constant added to the mini-batch variance for numerical stability.

2.3. Optimizers

Gradient Descent is one of the most popular algorithms to perform optimization on neural networks [Ruder 2016]. It simply changes the model parameters θ along the negative gradient direction, and this step is scaled by a learning rate η .

However, one of the weaknesses of this method is that the same learning rate η is applied to all parameters updates. For example, if the data is sparse, some feature dimensions will require larger updates on the gradient direction, whereas another features will only require a minimal update. With a fixed learning rate for all parameters, this kind of update is unfeasible. In order to address this problem, new optimization algorithms were proposed.

2.3.1. RMSProp

RMSProp [Tieleman and Hinton 2012] is a optimization algorithm quite similar to Adadelta [Zeiler 2012]. It divides the learning rate by an exponentially decaying average E of previous squared gradients g^2 , at time t . The authors suggest a decay constant γ to be set 0.9, while a good default value for the learning rate η is 0.01. The parameter update rule is given by

$$\begin{aligned} E[g^2]_t &= \gamma E[g^2]_{t-1} + (1 - \gamma)g_t^2 \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}}g_t \end{aligned} \quad (5)$$

2.3.2. Adam

Adam [Kingma and Ba 2014] is another optimization method that compute per-dimension parameters updates. The name *Adam* is derived from adaptive moment estimation. According to the authors, it is designed to combine the advantages of AdaGrad [Duchi et al. 2011], which works well with sparse gradients, and RMSProp [Tieleman and Hinton 2012] which works well in on-line and non-stationary settings.

In addition to storing an exponentially decaying average of past squared gradients v_t like Adadelta and RMSProp, Adam also keeps an exponentially decaying average of past gradients m_t , similar to momentum [Qian 1999]:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1)g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2)g_t^2 \end{aligned} \quad (6)$$

where m_t and v_t are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients, respectively. The authors indicated that these moment estimates are biased towards zero, which impairs the learning process. To avoid this, the authors proposed a bias correction, yielding the final Adam update rule:

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1 - \beta_1^T} \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^T} \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}}\hat{m}_t \end{aligned} \quad (7)$$

The default values for β_1 , β_2 and ϵ are 0.9, 0.999, and 10^{-8} , respectively.

2.3.3. Nadam (Nesterov momentum + Adam)

Nadam [Dozat 2016] incorporates Nesterov momentum [Nesterov 1983] into Adam [Kingma and Ba 2014]. The equations are almost the same, except for now we define a momentum decay schedule given by μ_t , applied to the gradient and to first momentum (the second momentum remains the same). Finally, the first momentum is modified and we have the final Nadam update rule (the another equations remain the same):

$$\begin{aligned}
 \mu_t &= \beta_1(1 - 0.5 * 0.96^{0.004t}) \\
 \hat{g}_t &= \frac{g_t}{1 - \prod_{i=1}^t \mu_i} \\
 \hat{m}_t &= \frac{m_t}{1 - \prod_{i=1}^{t+1} \mu_i} \\
 \bar{m}_t &= (1 - \mu_t)\hat{g}_t + \mu_{t+1}\hat{m}_t \\
 \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \bar{m}_t
 \end{aligned} \tag{8}$$

3. The original model

In our study, we focus on the original neural network model proposed by the authors [Bowman et al. 2015]. The first layer is the *sentence model layer* that produces a vector representation of each (premises and hypothesis) sentences, which is constructed without any exchange of information (word or phrase level) between these two sets of sentences. It was tested three sentence embedding models: *baseline sentence embedding model* which simply sums the embeddings of the words in each sentence; and two *sequence embedding models*: a *plain Recurrent Neural Network* (RNN) [Elman 1990] and a LSTM [Hochreiter and Schmidhuber 1997]. The word embeddings for all models are initialized with the 300-dimension reference GloVe vectors (840B token version) [Pennington et al. 2014].

The premise and hypothesis sentence vectors are mapped into a lower-dimensional space 100d (dimensions), using a additional tanh layer. Then they are concatenated and taken as input for a stack of three 200d layers, with tanh activation function. Finally a *softmax layer* provides the probability distribution of the sentence pair among the three possible classes (*entailment*, *contradiction* and *neutral*). All models are randomly initialized with standard techniques and trained using *Adadelta* [Zeiler 2012] until performance on the development step stops improving. A *L2 regularization* and *dropout* (which is a form of regularization that randomly remove layer units along with their connections) were applied to the inputs and outputs of the sentence embedding models though not to its internal connections [Bowman et al. 2015].

4. Model variants

Starting from the original model presented in section 3, we proposed 15 modifications, which are summarized on Table 1. The ? indicates that the correspondent parameters were not informed in [Bowman et al. 2015]. The main changes and techniques analyzed in our work were:

- On our experiments we used two pre-trained word vectors of different size, in order to check whether the word space dimension plays an important role on the final results:
 - A 100d version trained on the Wikipedia 2014 + Gigaword 5 corpus, with 6B tokens;
 - The 300d version trained on the Common Crawl corpus, with 840B tokens.
- On the sentence model layer, we made a few modifications:
 - We used the LSTM without dropout on its internal connections (like the original model) and without dropout on the sentence model layer's inputs and outputs, to check the dropout importance on the sentence model layer.
 - We tested the LSTM with *Variational Dropout* [Gal and Ghahramani 2016], with probability $p = 0.5$ on both \mathbf{z}_x and \mathbf{z}_h , and without dropout on the sentence model layer's inputs and outputs. This kind of dropout showed excellent results on language modeling tasks, outperforming the previous state-of-the-art model [Gal and Ghahramani 2016], and on sentiment analysis tasks, so we decided to check its performance on another NLP task.
 - We tested the RNN with standard weight matrix initializations. More precisely, we initialize the input and recurrent weight matrices with Glorot Uniform [Glorot and Bengio 2010] and Orthogonal [Saxe et al. 2013] initializers, respectively. Our objective is to check the performance of a default RNN on the NLI task.
 - We tested the RNN with the *Identity matrix* as initial recurrent weight matrix [Le et al. 2015]. This initialization improved the RNN performance on various tasks [Le et al. 2015], so we decided to check its performance on NLI task.
- The activation function that maps the premise and the hypothesis into a lower-dimensional space was changed to ReLU. The technical advantages of the ReLU over tanh activation function are that it does not involve expensive-to-compute functions, and it does not saturate [Goldberg 2015]. When an activation function saturates, the gradients at this region of the function are near zero, driving the entire gradient to zero. In this situation, the network can not optimize its parameters. The ReLU activation does not have these problems.
- We applied *batch normalization* on the output of the sentence model layer before concatenating the two sentences, and then we take the concatenated sentence as input for a stack of three 200d ReLU fully-connected plus Batch Normalization (FC + BatchNorm) layers, and finally a *softmax layer*. The batch normalization was originally applied on image classification models. We decided to check its performance on a NLP task.
- We tested different optimizers, namely: RMSProp [Tieleman and Hinton 2012], Adam [Kingma and Ba 2014] and Nadam [Dozat 2016], to check whether a different optimizer could improve the results. We also tested the learning rate changes suggested on [Ioffe and Szegedy 2015], to check whether this recommendation can be successfully applied to per-dimension optimization methods on a NLP task.
- We reduce L2 penalty on the weights, following [Ioffe and Szegedy 2015], to check its impact on the final results.

Tabela 1. Model Parameters

	dim.	optimizer	dropout	L2 penalty	learnig rate
sum of words	300	Adadelta	standard	?	?
1	100	Adadelta	no DP	4e-6	1.0
2	300	RMSProp	no DP	4e-6	0.001
3	300	RMSProp	no DP	4e-3	0.001
4	300	RMSProp	no DP	4e-6	0.045
5	300	Adam	no DP	4e-6	0.001
6	300	Nadam	no DP	4e-6	0.002
7	300	Adadelta	no DP	4e-6	1.0
RNN model	300	Adadelta	standard	?	?
8	300	Nadam	no DP	4e-6	0.002
9	300	RMSProp	no DP	4e-6	0.001
10	300	Nadam	no DP	4e-6	0.002
11	300	RMSProp	no DP	4e-6	0.001
LSTM model	300	Adadelta	standard	?	?
12	300	Nadam	no DP	4e-6	0.002
13	300	RMSProp	no DP	4e-6	0.001
14	300	Nadam	variational	4e-6	0.002
15	300	RMSProp	variational	4e-6	0.001

5. Modified Models

The *baseline sentence embedding model* [Bowman et al. 2015] was trained using a pre-trained 300-dimension GloVe version, containing more than 2 million words. The word space was fine-tuned as part of training. The model parameters were initialized using standard techniques and training using Adadelta. A L2 regularization was applied (the strength coefficient was not informed by the authors), and additionally applied dropout to inputs and outputs of the sentence embedding model. The learning rate used on Adadelta was not informed by the authors.

We proposed 7 modifications. The first one was upon the original sum of words model, and the other ones were upon the first modified model:

1. We used a pre-trained 100-dimension GloVe version [Pennington et al. 2014] (we did not modify the word vectors during training), expecting that decreasing the dimension will impair model performance. We tested the Adadelta optimizer, we applied a small L2 penalty of 4e-6 upon our FC+BatchNorm layers, originally applied to image classification tasks, and we removed dropout to inputs and outputs of the sentence embedding layer [Ioffe and Szegedy 2015].
2. We changed the first modified model (1), now using a pre-trained 300-dimension GloVe version and the RMSProp optimizer, to check whether a optimizer similar to Adadelta exhibits a similar performance.
3. On [Ioffe and Szegedy 2015], the authors reduced the weight regularization on their modified Inception model, and they achieved an accuracy improvement on the ImageNet Classification task [Russakovsky et al. 2015]. On all our previous models, we are using a reduced learning rate of 4e-6, assuming that this weak

regularization, along with the Batch Normalization layers, will exhibit a good performance, outperforming the original models. To check whether this recommendation (use a weak L2 penalty) is really necessary to achieve the best result, we tested a model that disregards the recommendation, setting the L2 penalty to $4e-3$ on this modified model.

4. We changed the model (2), increasing the learning rate to 0.045 (the original is 0.001), following another recommendation present in [Ioffe and Szegedy 2015].
5. According to [Ruder 2016], Adam [Kingma and Ba 2014] might be the best overall choice (his work did not consider Nadam [Dozat 2016]). In order to verify this statement, we changed model (2) optimizer to Adam, with default learning rate of 0.001.
6. According to [Dozat 2016], in most cases, like word analogy and language modeling tasks, the improvement of Nadam over Adam is fairly dramatic. We decided to check this statement on NLI task, modifying the model (2).
7. We changed the model (2), using Adadelta as optimizer. This model is the most similar to the original one, and the objective of this test is to check the dropout influence in final results.

The original model using RNN was trained under the same conditions as the original sum of words one. We proposed 4 modifications:

8. We used the 300-dimension GloVe version, without updating word vectors. We initialized the RNN input and recurrent weight matrices with Glorot Uniform [Glorot and Bengio 2010] and Orthogonal [Saxe et al. 2013] initializers, respectively (according to [Chollet et al. 2015]). We removed dropout from the model and we applied a small L2 penalty of $4e-6$ upon our FC+BatchNorm Layers [Ioffe and Szegedy 2015]. We tested this model using the Nadam optimizer with *gradient clipping* [Pascanu et al. 2013] (threshold=1.0), expecting that a more complex optimizer improve the model performance.
9. As we can see later on the Results section, the RMSProp optimizer exhibited the best results, so we decided to check the performance of this optimizer on the RNN and LSTM models. On the RNN, we modified the previous model (8), changing the optimizer to RMSProp.
10. We modified the weights initialization of the model (8), following [Le et al. 2015]. We initialize the recurrent weight matrix using the identity matrix scaled by 0.01, and the input weights were initialized with a random matrix, whose entries are sampled from a Gaussian distribution with mean of zero and standard deviation of 0.001, with ReLU activation function. This modifications were originally applied on the adding problem, designed to examine the power of recurrent models in learning long-term dependencies, MNIST [LeCun et al. 1998] digits classification when the 784 pixels are presented sequentially to the recurrent net, speech recognition and language modelling tasks. We expect that the property of keeping the errors constant offered by this initialization, along with an activation function that does not saturate, improve the model performance, and outperform the previous RNN models.
11. We modified the optimizer of the previous model (10), changing to RMSProp.

The original model using LSTM was trained under the same conditions as the original sum of words one. We proposed 4 modifications:

12. We used the 300-dimension GloVe version, without updating word vectors. We removed dropout from the whole model and we applied a small L2 penalty of $4e-6$ upon our FC+BatchNorm Layers [Ioffe and Szegedy 2015]. We tested this model using the Nadam optimizer, to check its performance, expecting that a more complex optimizer, along with the recommendations in [Ioffe and Szegedy 2015], improves the model performance.
13. As mentioned earlier on the RNN variants subsection, we tested the RMSProp on our LSTM models, modifying the previous model (12).
14. We modified the model (12), applying the variational dropout to it [Gal and Ghahramani 2016]. The variational dropout improved the standard LSTM performance on language modeling and sentiment analysis tasks, so we decided to apply this kind of regularization to the NLI task, to validate the hypothesis that a strong regularization on the input and recurrent connections can efficiently control overfitting, and so improve the LSTM performance [Gal and Ghahramani 2016].
15. We modified the optimizer of the previous model (14), changing to RMSProp.

All variants were implemented using Keras, [Chollet et al. 2015], a Python framework for neural networks and deep learning, and trained on a Tesla K80 GPU.

6. Results

On the Table 2, we present the results of our experiments. We observed that:

1. The results were a bit worse, what indicates that the additional information contained on the extra dimensions is important to the final accuracy of the model.
2. The performance was improved. The RMSProp update rule is very similar to the Adadelta one. Due to this, what can explain the performance improvement are the dropout absence along with the batch normalization on the fully-connected layers.
3. The results on the dev and test set were better with this stronger regularization, indicating that, on this specific task, setting a weak L2 weight regularization is not essential, in order to achieve a good performance.
4. This modification do not worked well, probably due to the learning rate increase, what indicates that the learning rate increase should be more carefully applied on optimizers that use a per-dimension update rule.
5. The Adam results were worse on this specific NLP task.
6. Nadam outperforms Adam, as expected, providing another example task where the Nesterov Momentum improves the optimizer performance.
7. The dropout absence improved the results. Another modification that can explain the results improvement is the ReLU activation function. As pointed out previously, the ReLU does not saturate, that is, the gradients are far from zero. Gradients near zero cause the vanishing gradient problem [Bengio et al. 1994], a common problem on networks with recurrent structures, which impairs the learning process. Due to this property, the ReLU helps improving the final results.
8. As expected, the model outperforms the original one.
9. RMSProp outperforms Nadam on the train and dev set, but not on the test set.
10. As pointed out by [Glorot and Bengio 2010], a proper weight initialization plays an important role on the final results, as we can see on Table 2, and on the Deep Learning field, in general.

11. One more time, RMSProp did not outperform Nadam on the test set.
12. As expected, the model outperforms the original one, as we can see on the Table 2. Due to its internal architecture, the LSTM was expected to provide the best performance, what really happened.
13. RMSProp outperforms Nadam on the train and validation set, but not on the test set, one more time.
14. As we can see on the Table 2, this model achieved the best overall performance, empirically validating the hypothesis.
15. The proposed modification improved the train and validation results, although it did not improve the accuracy of the test set.

Tabela 2. Experiment Results (on %)

Model	Train	Dev	Test
baseline sum of words model	79.3	-	75.3
1	74.10	72.48	72.23
2	82.33	76.88	76.52
3	79.69	77.58	77.50
4	74.38	32.97	32.90
5	78.86	72.47	72.48
6	79.01	76.70	76.31
7	79.52	76.81	76.48
baseline RNN model	73.1	-	72.2
8	78.95	74.32	74.58
9	82.61	74.69	74.47
10	83.26	75.86	75.83
11	83.73	74.41	74.08
baseline LSTM model	84.8	-	77.6
12	93.24	79.86	78.98
13	93.98	80.17	78.73
14	88.88	80.88	80.78
15	89.64	81.56	80.62

7. Conclusions

In this work, we analyzed some modifications applied to neural network-based models described on [Bowman et al. 2015] used in the NLI task. Among the proposed modifications, those which indicated a improved performance were: (i) a higher dimensional space; (ii) the absence of dropout on the sentence embedding layer and variational dropout applied to LSTM (improving the LSTM performance - exhibiting the best overall results); (iii) RNN initialization with the identity matrix outperformed the previous RNN models; (iv) activation function change on the layer that maps sentence into a lower-dimensional space; (v) batch normalization technique proved its effectiveness on a NLP task, and showing that dropout can be removed to a network with batch normalization layers, without loss performance; (vi) the use of more complex optimizers, like Adam and Nadam; (vii) more complex sentence embedding model structures really improves the model performance on the NLI task; (viii) the more complex optimizer (Nadam) exhibits the

best overall performance; (ix) improve the initialization of the model: this approach is validated on our task, as we can see comparing the performance of the original RNN model, the RNN following the standard weight initializations presented on [Chollet et al. 2015] and finally the RNN initialized with the Identity matrix and Gaussian distribution.

The objective of this work was not to propose a new state of the art for the inference task, outperforming [Wang et al. 2017], but to serve as a centralized source of information on neural networks and Deep Learning and the performance of its various algorithms and optimization techniques, to be used in future works.

8. Acknowledgments

The authors would like to thank the Laboratório de Combinatória e Computação Científica (LC3 - UFRJ), for providing Tesla K80 GPU, and so the feasibility of this study.

Referências

- Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.
- Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proc., EMNLP 2015, Lisbon, Portugal, Sept. 17-21, 2015*, pages 632–642.
- Chollet, F. et al. (2015). Keras. <https://github.com/fchollet/keras>.
- Cooijmans, T., Ballas, N., Laurent, C., and Courville, A. C. (2016). Recurrent batch normalization. *CoRR*, abs/1603.09025.
- Dozat, T. (2016). Incorporating nesterov momentum into adam. In *International Conference on Learning Representations (2016)*.
- Duchi, J. C., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179 – 211.
- Gal, Y. and Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In *NIPS*, pages 1019–1027.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feed-forward neural networks. In *Proc. of 13th Int. Conf. on Art. Int. and Stat., AISTATS 2010, Sardinia, Italy, May 13-15, 2010*, pages 249–256.
- Goldberg, Y. (2015). A primer on neural network models for natural language processing. *CoRR*, abs/1510.00726.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

- Le, Q. V., Jaitly, N., and Hinton, G. E. (2015). A simple way to initialize recurrent networks of rectified linear units. *CoRR*, abs/1504.00941.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Maccartney, B. (2009). *Natural Language Inference*. PhD thesis, Stanford, CA, USA. AAI3364139.
- Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pages 372–376.
- Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *ICML (3)*, volume 28 of *JMLR Proceedings*, pages 1310–1318. JMLR.org.
- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proc., EMNLP 2014, Oct. 25-29, 2014, Doha, Qatar*, pages 1532–1543.
- Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.
- Saxe, A. M., McClelland, J. L., and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *CoRR*, abs/1312.6120.
- Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. of Machine Learning Research*, 15(1):1929–1958.
- Tieleman, T. and Hinton, G. (2012). Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.
- Wang, Z., Hamza, W., and Florian, R. (2017). Bilateral multi-perspective matching for natural language sentences. In *IJCAI 2017*.
- Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.

Evaluating Rhetorical Arguments through Fuzzy Systems

Helio H. L. C. Monte-Alto^{1,2}, Mariela Morveli-Espinoza¹, Cesar A. Tacla¹

¹CPGEI - Federal University of Technology – Curitiba – PR – Brazil

²Federal University of Paraná – Palotina – PR – Brazil

heliohenrique@ufpr.br, morveli.espinoza@gmail.com, tacla@utfpr.edu.br

Abstract. *One of the challenges of rhetorical arguments is how to evaluate which argument is more suitable to send to an opponent in a specific situation. This work presents a fuzzy-based decision-making mechanism for agents to appropriately choose one argument among different rhetorical arguments. Some scenarios were developed and the results yielded by our system were then compared to opinions given by real people who answered a questionnaire based on these scenarios. The system is also compared to the state of the art of models proposed in the literature. The results show that the approach improves the mental model concerning the decisions of argumentative agents, as well as it makes possible to design agents adapted for different application scenarios, such as educational environments.*

1. Introduction

Persuasive negotiation involves negotiating using rhetorical arguments (such as threats, rewards, or appeals), which act as persuasive elements that aim to force or convince an opponent to accept a given proposal [Ramchurn et al. 2003]. This work is focused on threats and rewards. We propose ways of deciding among two or more rhetorical arguments by means of fuzzy systems, aiming to choose the best argument that will persuade an opponent to perform some task.

Rhetorical arguments can be used to enhance dialogues such as negotiation ones because the proponent believes that they are more likely to be accepted by the opponent, and therefore the backed proposal as well. In addition to negotiation, rhetorical arguments may be used in applications of behaviour change, in fields as education, personal finance, healthy life-styles and so on [Hunter 2016]. In these kinds of applications, the aim is to produce models of persuasion that can be used by computers to try to persuade humans. Thus, it is necessary to develop agents that are able to reason and interact with the users in order to support them in their tasks, such as doing a guided educational activity in an educational virtual environment, serious games or helping them in an online negotiation. In this context, agents should be able to generate rhetorical arguments and choose the most suitable one according to the application domain and the situation or context.

One way of choosing the most suitable rhetorical argument from a set of generated ones is by using their strength values. Some ways of calculating the strength of threats and rewards are proposed in the literature [Morveli-Espinoza et al. 2016b, Morveli-Espinoza et al. 2016a], but they do not provide flexibility to model the evaluation of arguments since they do not take into account the application domain and context. Thus, the main goal of this work is to propose a decision-making mechanism for choosing the most suitable rhetorical argument to be sent in a given situation.

Since the decision about what argument to choose has a considerable measure of uncertainty, this work proposes to model the decision-making process by means of fuzzy systems. Fuzzy systems are based on the fuzzy set theory [Zadeh 1965], which deals with the vague aspect of information. A fuzzy system consists of three parts: (i) the linguistic variables and the partition of the universe of discourse; (ii) a fuzzy rule base; and (iii) an inference mechanism.

In [Morveli-Espinoza et al. 2016b, Morveli-Espinoza et al. 2016a], the authors propose ways to generate threats and rewards and to calculate the strength of these arguments in order to allow the proponent to decide among them. For such calculation, some aspects are considered, such as: (i) the importance of the goal of the opponent; (ii) the state of the goal, which indicates how reachable it is; and (iii) the execution credibility level of the proponent from the point of view of the opponent. In this paper we present a fourth factor that can be relevant in the decision-making process: the willingness, or effort, considered by the opponent for executing the task proposed.

In the modelling of the linguistic variables and fuzzy partition, these four aspects for choosing among arguments must be considered. Moreover, the rule base has to be defined manually, initially, on the intuition. In order to validate and adjust the rules, we applied an online questionnaire presenting three scenarios involving the choice among threats or rewards. This aims to compare the answer given by our fuzzy system to the answers given by real people, resulting in an adjusted rule set for improved decision-making.

This research consists of the following steps: (I) implementing the fuzzy system, which can decide among different arguments; (II) elaborating scenarios as test cases for our system; (III) elaborating a questionnaire to be answered by real people about the scenarios; (IV) comparing the choices made by the system with the choices made by people; (V) adjusting the fuzzy rules to comply more accurately with the choices made by people; and (VI) comparing our fuzzy system to related works.

The next section discusses the problem in a more detailed way. Section 3 presents the scenarios proposed as test cases of the proposed system. Section 4 presents the simulations and results, as well as the comparisons with the questionnaire and with other models proposed in the literature. Finally, conclusions and future work are presented.

2. Problem Description

This work proposes a decision-making mechanism for choosing a suitable argument among a set of rhetorical ones taking advantage of fuzzy systems. Such arguments must be generated by another mechanism not covered in this work, but proposed in other works such as [Morveli-Espinoza et al. 2016b]. The agents are considered to be typical BDI (*belief-desire-intention*) agents, with their own goals and beliefs; however, covering the details of the agents' architecture is not in the scope of this paper. Therefore, the beliefs and goals used in the proposed decision-making model are given *a priori*.

As mentioned before, four aspects should be considered in order to evaluate an argument: (i) the opponent's goals, (ii) the goal's reachability, (iii) the proponent's credibility and (iv) the effort or willingness to perform the required task from the point of view of the opponent. Each one of these variables are inherently charged with uncertainty,

since they are measured based on partial perceptions of the environment and probably on different agents' mental models and beliefs.

In a formal way, let us represent an argument Arg as a tuple $Arg = \langle Prop, Op, Proposal, Goal_{Op} \rangle$, where $Prop$ refers to the *proponent*, Op refers to the *opponent*, $Proposal$ refers to the action proposed for the opponent to perform, and $Goal_{Op}$ is the *goal* of the *opponent* that is being threatened or rewarded. Then we can define the input variables for the system as $VarsIn = \langle Imp_{Goal}, Cred_{Prop}, Reach_{Goal}, Will_{Proposal} \rangle$, where Imp_{Goal} is the importance of the opponent's goal, $Cred_{Prop}$ is the proponent's credibility, $Reach_{Goal}$ is the reachability of the opponent's goal, and $Will_{Proposal}$ is the willingness of the opponent to perform the demanded task.

Given these variables, we can define the linguistic terms and universe partitions for each one of them. For the sake of simplicity, we define that all variables have equal linguistic terms and universe partitions. It is possible to adjust them one by one later. Thus, the linguistic terms for the variables are defined as: $\forall v \in VarsIn, \tau(v) = \{Low, Medium, High\}$.

The output variable, which represents the expected decision taken by the opponent given the proposed argument, is defined as d and the linguistic terms are defined as: $\tau(d) = \{Reject, Rebut, Accept\}$. *Reject* means that the argument is totally rejected, i.e., there is no chance that the opponent will accept it. *Rebut* means that the opponent will reject the argument at first, but it may still be persuaded with further argumentation. It can also be thought as an opportunity for the opponent to question, or refute, the argument proposed. *Accept* means that the opponent will accept the argument easily.

The universe partition for all variables, including the output variables, was initially defined as a uniform grid partition as plotted in Figure 1. However, this partition can be tweaked in order to try to achieve better results.

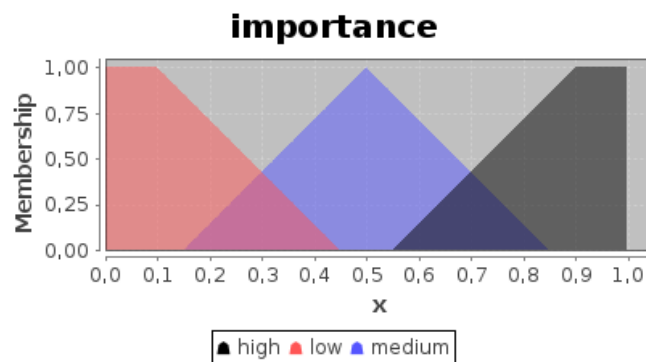


Figure 1. Uniform grid partition for the variables.

The initial rules were generated manually, based on intuition. In total, it was necessary to define 81 rules, since there are 4 input variables with 3 possible linguistic values each. These rules can also be adjusted in order to achieve better results, as presented in Section 4.3.

3. Scenarios and Questionnaire

This section presents the scenarios elaborated in order to test, validate and adjust our fuzzy system, as well as a questionnaire based on these scenarios. The first one was inspired on the threat scenario proposed in [Amgoud and Prade 2005], in which a mother tries to persuade her son to do his homework. The second one was inspired in the reward scenario proposed in [Morveli-Espinoza et al. 2016a], in which a boss tries to persuade his employee to work on weekends. The final scenario is an original one and related to one possible real application of agents with rhetorical arguments capabilities. It deals with both threats and rewards in an educational game scenario, in which a pedagogical agent (or intelligent tutor) tries to help and motivate the learner.

The next subsection will present each one of the scenarios, highlighting the four variables that are relevant for choosing a suitable argument.

3.1. Scenario 1: Threat

In this scenario, a mom asks her son to do his homework, but he refuses to do it. Therefore, she generates an argument to try to persuade him. She knows that the son has the following goals:

- **Goal 1:** “getting a Nintendo®”, and she knows that the son considers it **very important**. However, it is considered **little reachable** because the family currently does not have financial conditions to buy it;
- **Goal 2:** “going to the party”, and she knows that the son considers it **more or less important**. It is also **very reachable** because he has already been invited to it.

Given these two goals, the mom – the proponent in this case – generates two threats:

- **Threat 1:** “If you do not do your homework, you will not get a Nintendo®”;
- **Threat 2:** “If you do not do your homework, you will not go to the party”.

Two cases for credibility are considered in this scenario: (I) the mom has 50% credibility, i.e., she only fulfills about half of her threats; or (II) the mom has 100% credibility, i.e., she always fulfills her threats. Two cases are also considered for the willingness of the task: (I) the son hates to do his homework; or (II) the son considers doing his homework not so boring or difficult.

This scenario aims to emphasize the effects of the different strengths of the variables when choosing a threat.

3.2. Scenario 2: Reward

A boss wants to persuade his employee to work on weekends. Therefore, he must generate an argument to persuade him. He knows that the employee has the following goals:

- **Goal 1:** “visiting his parents”, who live far away. He considers it **more or less important**. However, he has back pain, which makes it more difficult to travel long distances, thus considering the goal **little reachable**;
- **Goal 2:** “repairing his car”. However, he likes to go to work by bus, thus considering the goal **little important**. Given that he is lacking a reasonable amount of money to pay for the repair, he considers it **more or less reachable**.

Given these two goals, the boss generated the following rewards:

- **Reward 1:** “If you work on weekends, you will have more holidays”;
- **Reward 2:** “If you work on weekends, I will pay you an extra salary”.

It is also considered that the boss fulfills his promises 100% of the time. The employee is more or less wishful to work on weekends.

3.3. Scenario 3: Threats and Rewards in an Educational Game

In an educational game, the learner has difficulties at the current stage. When finishing each stage, the learner usually wins 25 points. The learner has only three more chances to finish the stage. If he fails three more times, he will have to get back to the previous stage. He currently have 50 points and, if he manages to reach 100 points, he will win a badge, rising up in the ranking of the best learners.

An intelligent agent, which has control over the game and that tries to help and motivate the learner, realizes that the learner has difficulties at the current stage. He knows that the learner has the following goals:

- **Goal 1:** “Reaching 100 points”, being really cheerful about it (i.e., he considers it **very important**). However, because of the difficulty of the current stage, he considers it **more or less reachable**;
- **Goal 2:** “Not getting back to the previous stage”. He knows that if he fails three more times he will have to repeat the previous stage. Thus he considers this goal **very important**. But he also knows that if he fails only two more times or stop playing he will avoid it, thus he considers the goal **very reachable**.

Given the goals and the reasoning supporting them, the agent proposes the following two arguments (the first one being a reward and the second one being a threat):

- **Argument 1:** “If you manage to finish the stage now, you will win double points ($25 \times 2 = 50$), reaching 100 points”;
- **Argument 2:** “If you do not manage to finish the stage now, your chances will get reduced to 1 (i.e., you will have only one more chance to win)”.

The learner knows that the agent fulfills its threats/rewards 100% of the time. The learner considers the stage very hard to finish (low willingness).

3.4. Questionnaire

A questionnaire, based on an online form, was elaborated based on the three described scenarios. The aim of this questionnaire is to present the scenarios to real people and ask them to answer some question regarding the scenario, such as what argument they think the proponent should choose.

For the first scenario, the mother was presented initially with 50% credibility. Then five questions were made:

1. Which of the threats do you think the mother should choose? Justify it briefly;
2. Would the choice be different if the mother had 100% credibility?
3. Would the choice be different if the son hates to do his homework?

4. Considering that *Threat 1* is chosen, what do you think the son would answer (“Accept”, “Rebut” or “Reject”)? Justify it briefly;
5. Considering that *Threat 2* is chosen, what do you think the son would answer (“Accept”, “Rebut” or “Reject”)? Justify it briefly.

For the second scenario, three questions were made:

1. Which of the rewards do you think the boss should choose? Justify it briefly;
2. Considering that *Reward 1* is chosen, what do you think the employee would answer (“Accept”, “Rebut” or “Reject”)? Justify it briefly;
3. Considering that *Reward 2* is chosen, what do you think the employee would answer (“Accept”, “Rebut” or “Reject”)? Justify it briefly.

For the third scenario, three questions were made:

1. Which of the arguments do you think the agent should choose? Justify it briefly;
2. Considering that *Argument 1* is chosen, what do you think the learner would answer (“Accept”, “Rebut” or “Reject”)? Justify it briefly;
3. Considering that *Argument 2* is chosen, what do you think the learner would answer (“Accept”, “Rebut” or “Reject”)? Justify it briefly.

4. Simulations and Results

This section presents the results of the simulations performed with an implementation of the proposed fuzzy system, as well as the questionnaire results. A comparison is then made between the simulator and the questionnaire results.

4.1. Fuzzy System Results

For each scenario, a set of values for each variable was defined and used to feed the fuzzy system. These values are presented in Table 1. With these values, our fuzzy system yielded the decisions given in Table 2. This table also includes the trigger level of the rule triggered by the fuzzy system to yield the consequent. The trigger level is important to decide between the arguments when, for both, the system gives the same result – i.e., the trigger level is a tie breaker. It is important to note that the *t-norm* used was the *product* instead of *min*, mainly to avoid ties between trigger levels.

Table 1. Values assigned to each argument of the scenarios.

Scenarios:	Arguments:	Imp_{Goal}	$Cred_{Prop}$	$Reach_{Goal}$	$Will_{Proposal}$
Sc 1	<i>Threat 1</i>	0.9	0.5 or 1.0	0.2	1.0 or 0.2
	<i>Threat 2</i>	0.4		0.8	
Sc2	<i>Reward 1</i>	0.5	1.0	0.1	0.5
	<i>Reward 2</i>	0.1	1.0	0.5	0.5
Sc3	<i>Argument 1</i>	0.9	1.0	0.5	0.3
	<i>Argument 2</i>	0.9	1.0	0.9	0.3

Considering the probable responses given by the opponent as presented in Table 2, it is possible to define some simple rules for the proponent to decide between the arguments. A simple rule is to get the best fuzzy response: “Accept” is better than “Rebut” and “Rebut” is better than “Reject”. In case of tie, choose the argument whose triggered rule has the greater trigger level if the expected answer is “Accept”, and the lesser trigger

Table 2. Answers given by the fuzzy agent with the initial rules given the scenarios.

Scenarios	Arguments	Decision	Trigger level (<i>product</i>)
Sc 1	<i>Threat 1 100% Will 50% Cred</i>	Rebut	0.71
	<i>Threat 2 100% Will 50% Cred</i>	Accept	0.51
	<i>Threat 1 100% Will 100% Cred</i>	Accept	0.71
	<i>Threat 2 100% Will 100% Cred</i>	Accept	0.51
	<i>Threat 1 20% Will 50% Cred</i>	Rebut	0.51
	<i>Threat 2 20% Will 50% Cred</i>	Rebut	0.36
	<i>Threat 1 20% Will 100% Cred</i>	Rebut	0.51
	<i>Threat 2 20% Will 100% Cred</i>	Rebut	0.36
Sc 2	<i>Reward 1</i>	Accept	1.0
	<i>Reward 2</i>	Accept	1.0
Sc 3	<i>Argument 1</i>	Accept / Rebut*	0.42
	<i>Argument 2</i>	Accept	0.42

* There was a tie between two rules with the same trigger level.

level if the expected answer is “Reject”. For “Rebut”, the value in the x-axis (also called z in the fuzzy-related literature) given by defuzzification must also be considered, because a low pertinence for “Rebut” may mean that it leans either to “Accept” or to “Reject”. Thus, the lesser the x-axis is, the more it leans to “Accept” (see Figure 1). Given the results and these simple rules for the proponent to choose among the arguments, the arguments chosen for all the scenarios are presented in Table 3.

Table 3. Decision of what argument to propose and their expected response.

Scenarios:	Decision:
Sc1 100% Will 50% Cred	Threat 2 (expect “Accept”)
Sc1 100% Will 100% Cred	Threat 2 (expect “Accept”)
Sc1 20% Will 50% Cred	Threat 2 (expect “Rebut”)
Sc1 20% Will 100% Cred	Threat 2 (expect “Rebut”)
Sc2	Any (expect “Rebut”) ¹
Sc3	Argument 2 (expect “Accept”)

4.2. Questionnaire Results

The questionnaire was applied to students, professors and people in general. Since it does not require any previous knowledge or area of expertise, it was possible to open it to people unrelated to the academic field. In total, 22 people answered the questionnaire.

Figure 2 shows the answers to the questions I, II and III of Scenario 1. It is interesting to note that Threat 2 was overall considered a better choice than Threat 1. Many comments to justify the decision mentioned the fact that Threat 2 was better than Threat 1 because Goal 1 (“getting a Nintendo®”) was less reachable. This is evidence that, even when a goal is very important, the argument made of it may be weak if the goal is hardly reachable. In this aspect, we can see that our fuzzy agent was able to choose correctly for Scenario 1 in all cases.

In Figure 2 it is also interesting to note that, for Scenario 1, the increased credibility of the mom made Threat 1 just a little more interesting, whereas the decreased willingness of the task made Threat 1 considerably more interesting. This evidences that

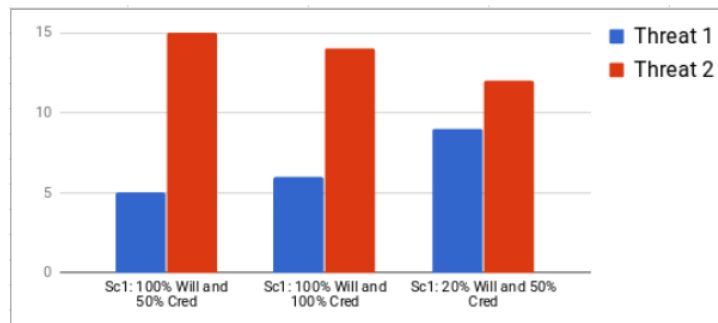


Figure 2. Answers for Scenario 1 cases.

willingness may be more relevant when the goal is very important, even if it is not very reachable.

Figure 3 shows the results for all three scenarios, considering the Scenario 1 with 100% willingness and 50% credibility. It is possible to note, in Scenario 2, that people consider the Reward 2 much more interesting than Reward 1, even when Reward 1 favoured a more important goal of the opponent. Among the many justifications, people argued that the employee could use money for other things besides fixing his car. Thus, it is interesting to realize that there are much more information that could be considered in this scenario, such as the *common belief* that money is important no matter the circumstances.

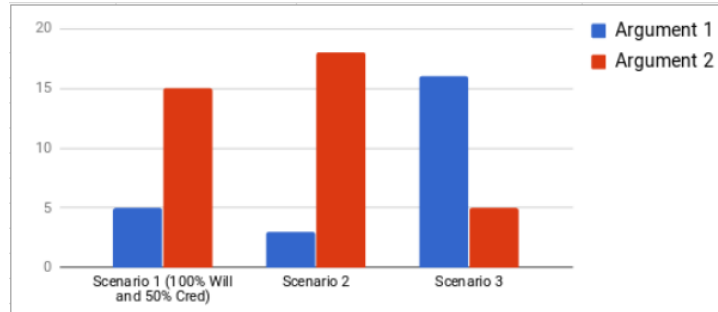


Figure 3. Answers for the first question of all scenarios.

For Scenario 3, Argument 1 was considered more interesting than Argument 2. Many respondents justified that it is better to motivate (by a reward) than to threaten, or that it is better to use a positive argument (the reward) instead of a negative one. One of the respondents answered interestingly that in a challenging environment, such as a game, threats become demotivating. Another person answered that Argument 1 gives more hope to the player, improving the player's self-esteem.

Another interesting comment states that if the player is cheerful to reach 100 points, getting back would only make him demotivated, causing him to stop playing. Such reasoning is very interesting because it appeals to emotional feelings (such as motivation, hope and self-esteem) that were not considered properly in the rules – such that the answers given by people are quite incompatible with our fuzzy system's results. It seems that the reward, although considered to have less persuasive power [Ramchurn et al. 2003], can be more interesting in an educational scenario, especially on a game that must keep

the user glazed and cheerful about going on.

One possible improvement to our initial fuzzy agent is to add an extra tie breaker when confronting a threat with a reward, making it possible to favour rewards for specific applications, such as an educational environment.

Figure 4 shows the responses for the last two questions of each scenario, concerning the answer that the opponent would give for each one of the arguments. It is interesting to note that most respondents answered that Threat 1 of Scenario 1 would be rebutted, whereas Threat 2 would be accepted. This agrees with our fuzzy agent decision. It is also interesting to note that Argument 2 of Scenario 3 has very divided opinions. This is due to the difficult reasoning involved on threatening the player, as already discussed.

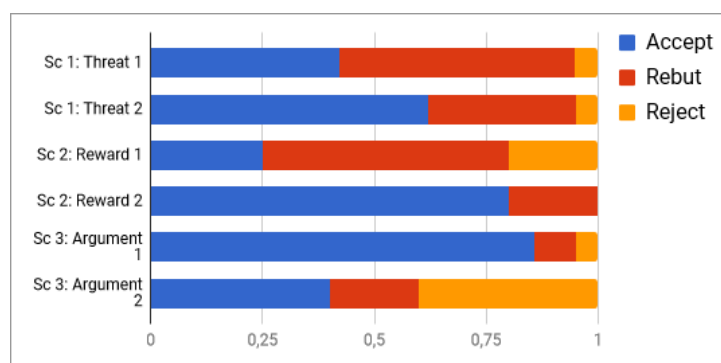


Figure 4. Answers for the last two questions of all scenarios.

4.3. Rules Adjustment

Based on the discrepancies observed between the answers given by the fuzzy system and the answers given by real people, we aim to adjust the fuzzy rules in a way that makes the fuzzy system answer accordingly. It is also necessary to change not only the rules directly related to the scenarios, but also similar rules. This was done by analyzing the rule that was triggered for each one of the answers, changing its consequent and analyzing the rules with similar antecedents.

Given that rewards and threats have different relevances in different application domains, it is proposed an intermediate rule to favour one over the other. In games and educational applications, it is considered that rewards are more appropriate. Thus, when there are two arguments – a threat and a reward – tied in such a scenario, the reward is chosen preferably.

Adjustments were made in rules triggered only in Scenario 2 and Scenario 3, since the results of the fuzzy agent for Scenario 1 agreed with the answers given by people. For Scenario 2, it is expected that Reward 2 should be chosen instead of Reward 1. The rule triggered by Reward 2 is the following:

$$\begin{aligned}
 & (Cred_{Prop} = High) \wedge (Imp_{Goal} = Low) \wedge \\
 & (Reach_{Goal} = Medium) \wedge (Will_{Proposal} = Medium) \\
 & \implies Decision_{Op} = Rebut
 \end{aligned} \tag{1}$$

Since this argument is preferable and it is reasonable to consider that a reward with high credibility of the proponent and medium reachability and willingness could be accepted, the consequent of this rule was changed to *Accept*. Rules with similar antecedents were also revised in order to find more candidate rules to be changed.

For Scenario 3, Argument 1, which should be accepted and preferred to Argument 2, had the following triggered rule:

$$\begin{aligned} Cred_{Prop} = High \wedge Imp_{Goal} = High \wedge \\ Reach_{Goal} = Medium \wedge Will_{Proposal} = Low \\ \implies Decision_{Op} = Rebut \end{aligned} \quad (2)$$

Since this argument is preferable and it is reasonable to consider that a reward with high importance, high credibility and medium reachability, even with a low willingness to complete the proposed task, could be accepted, the consequent of this rule was changed to *Accept*. Rules with similar antecedents were also revised in order to find more candidate rules to be changed.

Given these rule adjustments, the following changes were reflected in the results:

- For Scenario 2, the rules were changed so that Reward 2 results in an “Accept” consequence, causing it to be chosen since Goal 2 is more reachable.
- For Scenario 3, the rules were changed so that both arguments results in an “Accept” consequence, but the argument chosen by the agent is Argument 1 since it must favour rewards as a pedagogical agent.

It is possible that more adjustments in the rules must be made in order to make the agent even more intelligent in its decisions. More scenarios are necessary and will be developed in future works.

4.4. Related Works and Comparisons

Ramchurn et al. [Ramchurn et al. 2003] present a framework to describe persuasive negotiation using rhetorical arguments, considering the three kinds of rhetorical arguments: threats, rewards and appeals. These kinds of arguments have been studied in terms of speech acts in [Ramchurn et al. 2003] and in logical terms in [Amgoud and Prade 2005, Amgoud and Prade 2006]. Studies have also concentrated on the issue of generating and evaluating the strength of the arguments and how they can be defeated [Amgoud and Prade 2005, Morveli-Espinoza et al. 2016b].

It may be interesting to compare the results issued with other similar models using our scenarios. In [Morveli-Espinoza et al. 2016a] it is presented a formula to calculate the strength of rewards, and in [Morveli-Espinoza et al. 2016b] it is presented a similar formula to calculate the strength of threats. Actually, one of the formulas considers only the importance and reachability, which is called *Basic Strength*. In these works the reachability is modeled based on the belief-based goal processing (BBGP) model proposed in [Castelfranchi and Paglieri 2007], who propose 4 possible states of the goals of an agent: (1) active, (2) pursuable, (3) chosen and (4) executable. The closer to the executable state, the closer to become executable, thus the more reachable the goal is.

Given this modeling, the formula for the Basic Strength is given by:

$$ST_{basic}(arg) = \frac{(\frac{State(go_j)}{num_states} + Importance(go_j))}{2} \quad (3)$$

The *num_states* refers to the 4 possible states, and the *State(go_j)* refers to the current state of the goal – e.g., if the goal is pursuable, then *State(go_j)* = 2, and this divided by *num_states* = 4 gives 0.5 as result.

The credibility of the proponent can be considered by means of the *Combined Strength* formula, given by:

$$ST_{comb}(arg) = ST_{basic}(arg) \times Level_Exec_{arg}(Op) \quad (4)$$

Here, *Level_Exec_{arg}(Op)* ∈ [0, 1], and denotes the credibility level of an opponent *Op*.

Given the values presented in Table 1 for each one of the scenarios, Table 4 presents the results of the application of these formulas to each argument.

Table 4. Answers given based on the calculation of threats and rewards proposed by [Morveli-Espinoza et al. 2016b, Morveli-Espinoza et al. 2016a].

Scenarios:	Arguments:	Basic strength	Combined strength	Chosen?
Sc 1	<i>Threat 1 100% Will 50% Cred</i>	0,2	0,1	No
	<i>Threat 2 100% Will 50% Cred</i>	0,4	0,2	Yes
	<i>Threat 1 100% Will 100% Cred</i>	0,2	0,2	No
	<i>Threat 2 100% Will 100% Cred</i>	0,4	0,4	Yes
	<i>Threat 1 20% Will 50% Cred</i>	0,2	0,1	No
	<i>Threat 2 20% Will 50% Cred</i>	0,4	0,2	Yes
	<i>Threat 1 20% Will 100% Cred</i>	0,2	0,2	No
	<i>Threat 2 20% Will 100% Cred</i>	0,4	0,4	Yes
Sc 2	<i>Reward 1</i>	0,1	0,1	Yes*
	<i>Reward 2</i>	0,1	0,1	Yes*
Sc 3	<i>Argument 1</i>	0,5	0,5	No
	<i>Argument 2</i>	0,9	0,9	Yes

It is possible to note that, for all scenarios, this model gave very similar results in comparison to our model with the initial rules.

However, the fuzzy model has the advantage that it enables the adjustment of rules and even of the universe partition, which also may influence the results. This way, the fuzzy system allows to create agents with different mental models. For example, it makes it possible to create more naive agents or more skeptical agents by tweaking their fuzzy rules.

One disadvantage of the fuzzy model is the great amount of rules needed. With just 4 variables and 3 linguistic values, 3⁴ = 81 rules are needed. If we use more linguistic values, which could be interesting for greater precision, this number of rules increases by a large amount. For example, for 5 linguistic values it increases to 5⁴ = 625 rules, making it even more difficult to define the rules manually.

Conclusions and Future Work

It is interesting to realize the potential of fuzzy systems for decision-making involving rhetorical arguments. The main advantage over other methods, such as calculating the strength of arguments, is that the fuzzy rules are adaptable, i.e., it is possible to change the mental model and behaviour of an agent by tweaking its fuzzy rules. This way, it is possible to create agents with different characteristics that are more suitable for different scenarios. For example, a pedagogical agent in a serious game may favour rewards instead of threats, since it is often better to propose positive arguments to motivate the player. In the other hand, such strategy may not be well suited for a negotiation system scenario, in which agents might better be modelled as more skeptical and aggressive agents.

For future works it would be interesting to consider the other type of rhetorical argument not considered in this work: appeals. This remaining kind of argument may be interesting to improve the mental model of agents. We also intend to implement an example of a serious game with a pedagogical agent similar to that presented in Scenario 3. This way, it would be possible to validate this kind of environment as well as the effectiveness of the argumentation capabilities of the agents in an educational scenario. Another subject for further research is considering other contextual information and common beliefs, such as the importance of money mentioned by respondents of the questionnaire for Scenario 2. Finally, it would be interesting to discover better ways to generate the fuzzy rules for the agents, in order to reduce the effort needed to create and adjust the rules.

References

- Amgoud, L. and Prade, H. (2005). Handling threats, rewards, and explanatory arguments in a unified setting. *International Journal of Intelligent Systems*, 20(12):1195–1218.
- Amgoud, L. and Prade, H. (2006). Formal handling of threats and rewards in a negotiation dialogue. In Parsons, S., Maudet, N., Moraitis, P., and Rahwan, I., editors, *Argumentation in Multi-Agent Systems: Second International Workshop, ArgMAS 2005 Utrecht, The Netherlands, July 26, 2005 Revised Selected and Invited Papers*, pages 88–103. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Castelfranchi, C. and Paglieri, F. (2007). The role of beliefs in goal dynamics: prolegomena to a constructive theory of intentions. *Synthese*, 155(2):237–263.
- Hunter, A. (2016). Computational persuasion with applications in behaviour change. In *Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016.*, pages 5–18.
- Morveli-Espinoza, M. M., Possebom, A. T., and Tacla, C. A. (2016a). Constructing and calculating the strength of rewards. In *Proceedings of XIII Encontro Nacional de Inteligencia Artificial e Computacional (SBC ENIAC-2016)*, pages 61–72.
- Morveli-Espinoza, M. M., Possebom, A. T., and Tacla, C. A. (2016b). Construction and strength calculation of threats. In *Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016.*, pages 403–410.
- Ramchurn, S. D., Jennings, N. R., and Sierra, C. (2003). Persuasive negotiation for autonomous agents: A rhetorical approach. In *IJCAI Workshop on Computational Models of Natural Argument. Agosto-2003 Acapulco*, pages 9–17. IJCAI, IJCAI.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3):338–353.

Bayesian Sets for Binary Item Recommendation Systems

Diorge Brognara¹
Estevam R. Hruschka Jr.¹

¹Computing Department — Federal University of São Carlos

Abstract. *Set expansion algorithms aim for semi-supervised expansion of a given set of items. The application of such algorithms for recommender systems task is somewhat discussed in the literature, but the main goal is towards other fields such as extraction of named entities. We present the results of applying one such algorithm, the Bayesian Sets model, to the task of recommender systems using a binary feedback input, objectively comparing it to traditional algorithms for the task. The experiment indicates that the model has good precision and scalability.*

Resumo. *Algoritmos de expansão de conjuntos visam, de forma semissupervisionada, expandir um determinado conjunto de itens. A aplicação desses algoritmos para tarefas de recomendação é pouco discutida na literatura, em favor de outras aplicações como extração de entidades nomeadas. Apresentamos os resultados da aplicação de um algoritmo de expansão de conjuntos, o modelo de Conjuntos Bayesianos, para a tarefa de um sistema de recomendação com entrada binária, comparando objetivamente a algoritmos tradicionais para esta tarefa. O experimento indica que o modelo tem boa precisão e escalabilidade.*

1. Introduction

Recommender Systems (RSs) are traditionally used for recommending new items to ease the burden of choice to an user, in a customized fashion [Ricci et al. 2011]. Examples of such systems includes *e-commerce* shops, media content providers (videos, movies, songs), and advertisements managers. RSs are getting a lot of attention in the industry since the second half of the 90's [Adomavicius and Tuzhilin 2005], due to the rise of popularity of services such as *Amazon.com* [Linden et al. 2003], *Youtube*, *Netflix* [Ricci et al. 2011], and the competition *Netflix Prize* between 2006 and 2009, won by the work described in [Töscher et al. 2009], that was never used in practice due to implementation costs [Amatriain and Basilico 2012].

There are two main categories to classify a RS, the content-based filters and collaborative filters, both using items previously “used” (bought, seen) by the user [Adomavicius and Tuzhilin 2005]. In content-based filters, the system recommends items with similar characteristics to items previously used. Collaborative filters, on the other hand, use the similarity between behaviors of utilization, considering the usage history of all users, in an attempt to model user profiles [Jafarkarimi et al. 2012]. It's possible to mix both approaches to create hybrid filters [Chu and Tsai 2017].

A family of algorithms used to solve a class of semi-supervised problems are the set expansion algorithms. These methods aim to, given a small *seed* set, include new elements in this set in a sensible manner, such as keeping the maximum similarity

between the set elements [Zhou 2016]. A pioneer set expansion algorithm is the Bayesian Sets (BS), a probabilistic approach to rank every element using a confidence value for it pertaining to the seed set [Ghahramani and Heller 2006].

In this work, we explore many different collaborative filters using the MovieLens [Harper and Konstan 2016] and Jester [Goldberg et al. 2001] datasets, including baseline, traditional and more recent models largely used for RSs, and also, we propose a BS model, evaluating and comparing the performance of the different models, using binary feedback input. In Section 2 we introduce related approaches for solving the binary feedback RS task, as well as other works using set expansion algorithms. Section 3 presents the data and libraries used in the experiments of this paper, as well as the experiments conduction and results. Finally, in Section 4 we discuss the takeaways of the applicability of BS for the binary feedback RS task.

2. Related Works

There has been a similar comparison of traditional collaborative filters [Breese et al. 1998], including using the MovieLens dataset to look for optimal hyperparameters settings in [Herlocker et al. 1999], but such comparison focuses on nearest neighbors approaches only. In [Sarwar et al. 2000] the *Singular Value Decomposition* method is used to reduce the dimension before the nearest neighbors search. More recent works present the matrix factorization technique to improve the scalability of RSs ([Frolov and Oseledets 2017], [Koren et al. 2009], [Rendle and Schmidt-Thieme 2008]).

Previous methods focus on predicting the rating of each pair (*user, item*). Some applications may not have a rating, but instead only positive feedback, such as “the user has bought this item”, thus making it a binary feedback. The discussion of this problem led to methods such as Bayesian Personalized Ranking [Rendle et al. 2009], Soft Margin Matrix Factorization [Rendle et al. 2009], and Sparse Linear Methods [Ning and Karypis 2011].

In the seminal work for BS ([Ghahramani and Heller 2006]), there’s mention of the applicability for RSs, but it is not deeply explored. The main applicability of set expansion algorithms is for the named entities extraction ([Paşca 2007], [Wang and Cohen 2008], [Rong et al. 2016], [Zhou 2016]). [Zhang et al. 2016] presents a set expansion algorithm for the extraction of relevant features.

3. Experiments and results

Three datasets were used: the Jester dataset [Goldberg et al. 2001], and two versions of the MovieLens dataset [Harper and Konstan 2016], named MovieLens100k (with 100,000 ratings) and MovieLens1M (with 1 million ratings). An overview of the datasets can be found in table 1, where density is $\frac{|Ratings|}{|Users| * |Items|}$. Jester is a dataset of joke reviews, generated from a website¹ where users are shown a joke and immediately asked to rate it between “Less Funny” and “More Funny”, in a continuous scale. The MovieLens

¹eigentaste.berkeley.edu

website² provides “non-commercial, personalized movie recommendations”, in exchange of anonymous usage data for research purposes.

Dataset	Ratings	Unique users	Unique items	Density
MovieLens 100k	100,000	943	1,682	6.3%
MovieLens 1M	1,000,209	6,040	3,706	4.4%
Jester	1,810,455	24,983	100	72.4%

Table 1. Datasets overview

The data used was lightly cleaned. Both datasets use ratings, ranging from -10 to 10 for Jester and 1 to 5 for MovieLens. Since we’re interested in binary item recommendation instead of rating prediction, the ratings were binarized, with values greater than 0 considered positive for Jester, and values greater than or equal 3 (the median) considered positive for MovieLens. Jester also had some users with very few positive ratings, we dropped users with four or less positive ratings (195 out of 24983 users). This threshold is chosen because of the validation method Keep 4, explained below. To avoid overfitting of the models, all the datasets were split using the K-Fold technique, with 10 folds, splitting by users (i.e. 90% users for training and 10% users for testing).

The traditional models used were the Most Popular (MP, which simply recommends the most popular items), Item KNN (KNN), Bayesian Personalized Ranking Matrix Factorization (BPRMF), and the Soft Margin Matrix Factorization (SMRMF). The implementation used was provided by the MyMediaLite library [Gantner et al. 2011], and the hyperparameters were not tuned, using the sensible defaults instead to avoid overfitting the models. The BS model was implemented using the Numpy and Scipy libraries for Python, and the hyperparameters were not tuned either, using the same suggestion of $C = 2$ as [Ghahramani and Heller 2006]. All experiments were run on the same host, using a Intel i7-6700HQ processor with 16GB RAM.

There’s heavy discussion about evaluation of RSs (see [Shani and Gunawardana 2011]). For this offline experiment (i.e. tested on the data itself and not on real users), we are using the precision and coverage metrics, measured at five, ten and twenty recommendations. To test the precision metric, two strategies of “hiding” the information were used, one where half of the positive ratings are hidden and the other half is provided to the algorithm (“Half-Hidden”), and another where only four of the positives ratings are kept and the rest is hidden (“Keep 4”). This choice is to simulate the Cold Start problem (see [Ricci et al. 2011], [Schein et al. 2002]), since every application of RS is susceptible to this effect. Thus, the Keep 4 method simulates the start of the RS application, by providing the recommender with less information but allowing more possible correct hits, while the Half-Hidden method is more similar to well-established RSs, providing more information and less possible correct hits. Since these are validation methods, they are applied only on the test data and not on training data.

Figures 1 and 2 show the precision of each recommender for the Half-Hidden method, for datasets MovieLens100k and MovieLens1M, respectively. The black ticks at the top of the bars show the 5% and 95% confidence intervals. In a similar fashion,

²<https://movielens.org>

figures 3 and 4 show the precision for the Keep 4 method in datasets MovieLens100k and MovieLens1M.

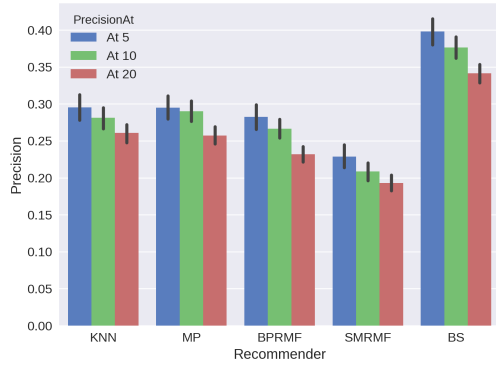


Figure 1. Precision for Half-Hidden in MovieLens100k

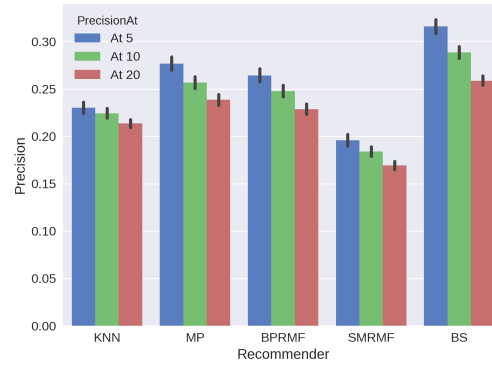


Figure 2. Precision for Half-Hidden in MovieLens1M

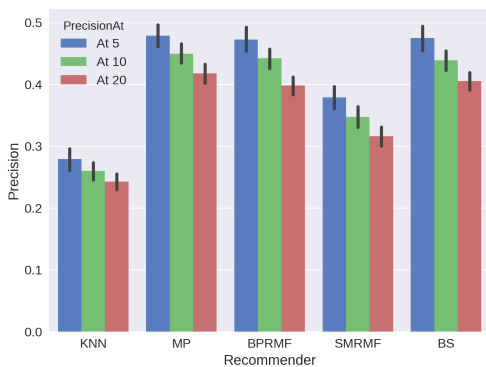


Figure 3. Precision for Keep 4 in MovieLens100k

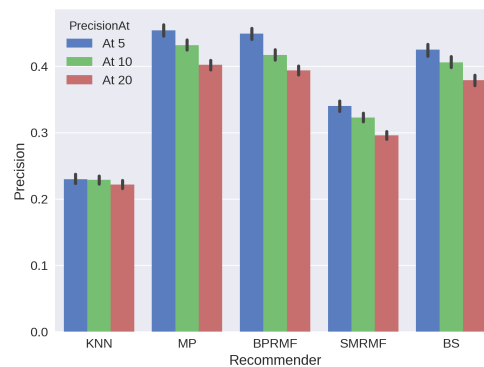


Figure 4. Precision for Keep 4 in MovieLens1M

For the Jester dataset, every recommender achieved precision close to 1.0, for both validation methods, which can be explained by the very high density of the dataset. In the much less dense MovieLens datasets, the Most Popular, BPR MF and BS models have better precision when given a query of length four, and the BS model has a much better result for queries of longer length.

As for the coverage analysis, figures 5 and 6 show the coverage of each recommender for the dataset MovieLens100k, when validated with the Half-Hidden and Keep 4 methods, respectively. Similarly, figures 7 and 8 show the coverage for MovieLens1M, and figures 9 and 10 for Jester. The KNN method had better results in every test, followed by the BS model. The two different validation methods influence the absolute coverage values, but the relation between the different recommenders stay relatively the same when comparing the Half-Hidden and Keep 4 results for the same dataset. The BPR MF model had just barely better coverage than the Most Popular approach, which is supposed to have the minimum possible coverage, since it recommends a fixed set of items.

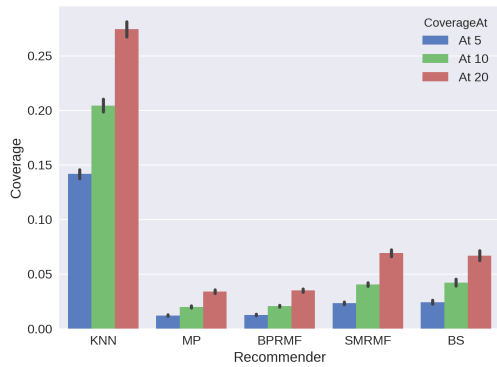


Figure 5. Coverage for Half-Hidden in MovieLens100k

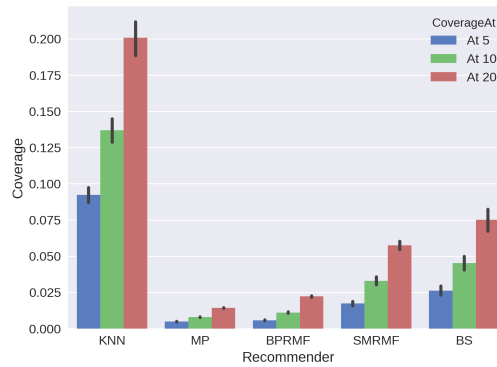


Figure 6. Coverage for Keep 4 in MovieLens100k

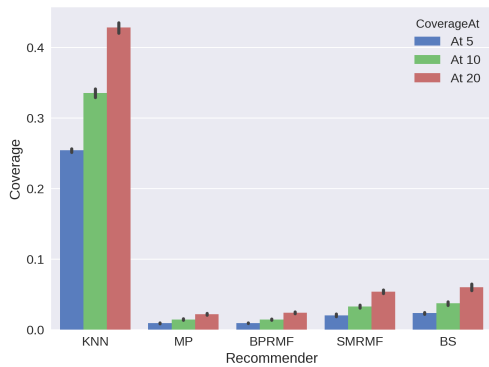


Figure 7. Coverage for Half-Hidden in MovieLens1M

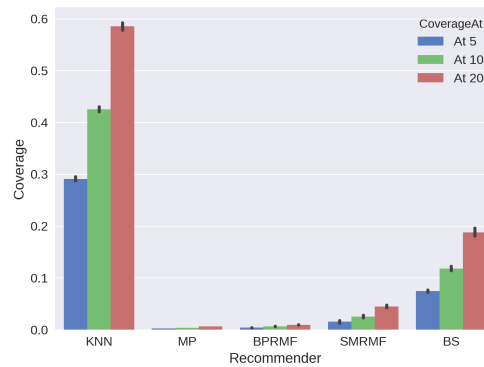


Figure 8. Coverage for Keep 4 in MovieLens1M

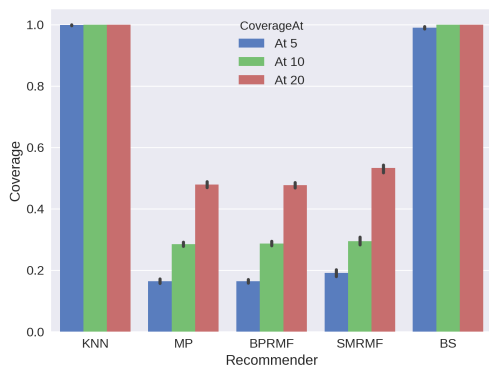


Figure 9. Coverage for Half-Hidden in Jester

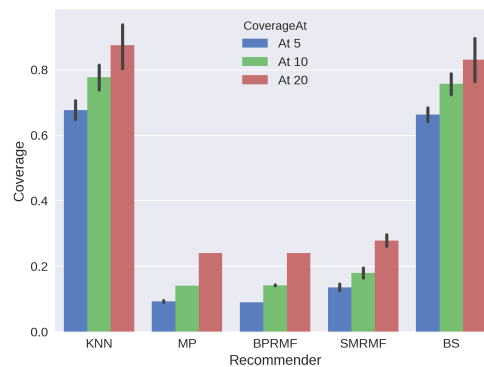


Figure 10. Coverage for Keep 4 in Jester

For the time performance of each model, figure 11 shows the elapsed time to train the models, with the Most Popular and Bayesian Sets taking an insignificant amount of time, as expected, since Most Popular is a very simple counting algorithm, and BS

is mostly a lazy model. The MF methods take less time than KNN on the MovieLens datasets, but do not scale well with the density of the Jester dataset. KNN taking a non-insignificant amount of time indicates MyMediaLite's implementation is not lazy, instead precomputing the model.

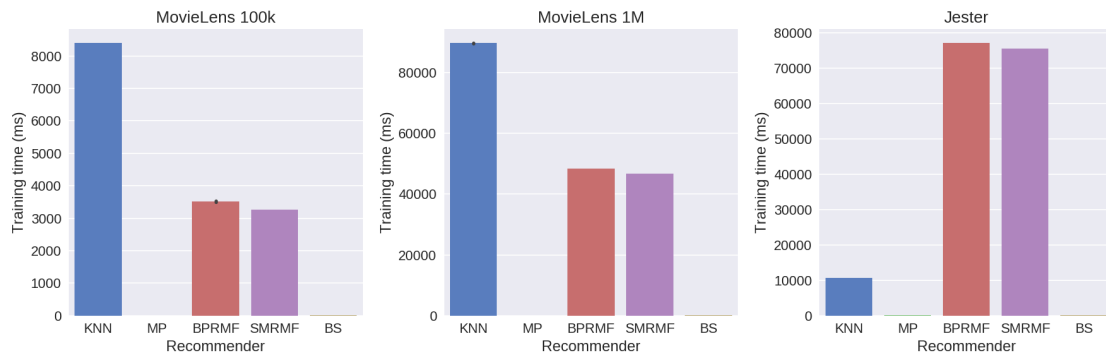


Figure 11. Elapsed time for training the models

In figure 12, showing the time elapsed to test each model for each fold of the split, a few details are worth commenting. The Most Popular method takes a non-insignificant amount of time, which scales with the density of the input. This signals either the MyMediaLite library or our usage of it has some strong performance overhead upon recommending new items. Thus, if we disregard such overhead, all the methods, except for KNN, take an actual insignificant amount of time to recommend new items. The KNN method takes a very long time for evaluating new recommendations, even with the already long time for training the precomputed model.

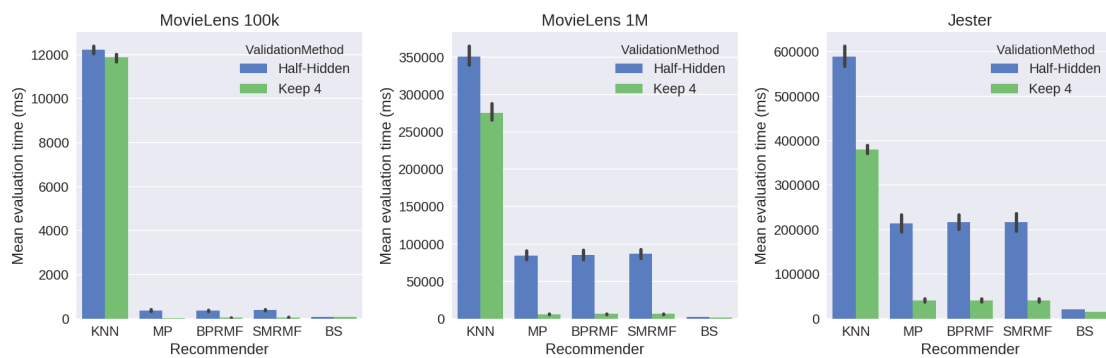


Figure 12. Elapsed time for evaluating the models

There was also an attempt to do the same analysis for the MovieLens 20M dataset, containing twenty million reviews. However, after trying different implementations of the traditional methods, all failed to produce results with the available RAM memory and processing power. Nonetheless, the Bayesian Sets model was able to produce results with minimal memory footprint in a timely manner. Unfortunately we found no way of comparing the results obtained. Using a simple 25/75 split for train and test data, the model obtained 28.04% mean precision, 22.95% coverage, training the model in less than one second, and evaluating the test set in just 1320 seconds (22 minutes).

4. Conclusion

Set expansion can be seen as the task of expanding a partial set of *seed* instances of a concept into a more complete set of them. Therefore, such a task seems to fit very well in *recommendation systems* approaches. In spite of the intuitive suitability of coupling together *set expansion* algorithms and *recommendation systems*, it is not well explored in the literature, yet. In this work, we explore the use of different well-known collaborative filters in a *recommendation system* setting and empirically compare the results obtained when using the proposed idea of *set expansion* approach to the same task. We use Bayesian Sets algorithm as our set expansion model.

The empirical results revealed that Bayesian Sets model achieves good precision, while keeping great coverage and scalability, when compared to other binary feedback algorithms. It corroborates the intuitive idea that *set expansion* models can play an important role in recommendation systems. Bayesian Sets ignores, however, the information provided by the actual rating.

In future works we intend to explore an interesting approach (proposed in [Verma and Hruschka Jr. 2012]) of *Coupled Bayesian Sets* (CBS). In addition, we want to investigate how BS, as well as CBS fares against models that use information provided by the actual rating. It is also worth exploring if these results stay consistent with changes to hyperparameters of the algorithms. Overall, in our experiments, Bayesian Sets have shown to be a good model for very sparse binary feedback, that can possibly be used in real-time requirements.

References

- Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.
- Amatriain, X. and Basilico, J. (2012). Netflix recommendations: Beyond the 5 stars (part 1).
- Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI'98*, pages 43–52, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Chu, W.-T. and Tsai, Y.-L. (2017). A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web*, pages 1–19.
- Frolov, E. and Oseledets, I. (2017). Tensor methods and recommender systems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(3).
- Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2011). MyMediaLite: A free recommender system library. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*.
- Ghahramani, Z. and Heller, K. A. (2006). Bayesian sets. In *Advances in neural information processing systems*, pages 435–442.
- Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, 4(2):133–151.

- Harper, F. M. and Konstan, J. A. (2016). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19.
- Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 230–237, New York, NY, USA. ACM.
- Jafarkarimi, H., Sim, A. T. H., and Saadatdoost, R. (2012). A naïve recommendation model for large databases. 2(3).
- Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8).
- Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.
- Ning, X. and Karypis, G. (2011). Slim: Sparse linear methods for top-n recommender systems. In *ICDM 2011*.
- Paşca, M. (2007). Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 683–690, New York, NY, USA. ACM.
- Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *UAI 2009*.
- Rendle, S. and Schmidt-Thieme, L. (2008). Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 251–258. ACM.
- Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to Recommender Systems Handbook*, pages 1–35. Springer Science+Business Media.
- Rong, X., Chen, Z., Mei, Q., and Adar, E. (2016). Egoset: Exploiting word ego-networks and user-generated ontology for multifaceted set expansion. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 645–654. ACM.
- Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2Nd ACM Conference on Electronic Commerce*, EC '00, pages 158–167, New York, NY, USA. ACM.
- Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 253–260, New York, NY, USA. ACM.
- Shani, G. and Gunawardana, A. (2011). *Evaluating Recommendation Systems*, pages 257–298. Springer Science+Business Media.
- Töscher, A., Jahrer, M., and Bell, R. M. (2009). The bigchaos solution to the netflix grand prize.

- Verma, S. and Hruschka Jr., E. R. (2012). Coupled bayesian sets algorithm for semi-supervised learning and information extraction. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2012)*, Bristol, UK. Association for Computing Machinery.
- Wang, R. C. and Cohen, W. W. (2008). Iterative set expansion of named entities using the web. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 1091–1096. IEEE.
- Zhang, Z., Sun, L., and Han, X. (2016). A joint model for entity set expansion and attribute extraction from web search queries.
- Zhou, K. (2016). Offline strategies for online set expansion. Master's thesis, University of Alberta.

Aplicando a Mineração de Opiniões na Avaliação de Relatórios de Pesquisas Científicas

João Pedro Santos Rodrigues¹, Zacarias Curi¹, Mariza Miola Dosciatti²,
Emerson Cabrera Paraiso¹

¹Laboratório de Sistemas Inteligentes - LASIN
Programa de Pós-Graduação em Informática – PPGIA
Pontifícia Universidade Católica do Paraná (PUCPR) - Curitiba, PR – Brasil

²Departamento Acadêmico de Informática - DAINF
Universidade Tecnológica Federal do Paraná (UTFPR) - Pato Branco, PR - Brasil

jpsanr@gmail.com, {zacarias, mariza.dosciatti, paraiso}@ppgia.pucpr.br

Abstract. *In this article we apply Opinion Mining to evaluate academic reviews written in Brazilian Portuguese. The method, based on supervised learning, is able to identify the polarity of the evaluation text of a research project. The corpus used to evaluate the method is compounded by 1,119 reviews of undergraduate research projects. Each project has a review and a final grade. It is expected that approved projects have positive reviews (in terms of text polarity). Experiments show that the method achieved an accuracy rate of 86% in identifying the polarity of a review.*

Resumo. *Neste artigo apresentamos um método, utilizando a mineração de opiniões, para a avaliação automática de pareceres relacionados a pesquisas desenvolvidas por estudantes de iniciação científica escritos em português do Brasil. O método foi implementado em uma ferramenta de auxílio aos avaliadores. O método, baseado na aprendizagem supervisionada, é capaz de identificar a polaridade do texto de avaliação de um trabalho. A polaridade é utilizada na verificação da nota final atribuída a avaliação. Em experimentos realizados com uma base de 1.119 avaliações, o método obteve uma taxa de acerto de 86%.*

1. Introdução

As avaliações são fundamentais para o desenvolvimento das competências de estudantes. Essa tarefa deve ser mais ampla que a simples atribuição de notas ou classificações, uma avaliação autêntica deve indicar erros importantes na ótica da construção das competências e contribuir para o desenvolvimento dos estudantes [Perrenoud et al. 2009]. A correta execução desse processo permite aos estudantes uma análise do que foi desenvolvido, gerando um autoconceito e uma reflexão em relação ao seu próprio desempenho [Marques et al. 2006].

Nas modalidades de ensino atuais, principalmente no ensino a distância (EAD), a clareza e objetividade nos comentários das avaliações torna-se ainda mais relevante, devido ao distanciamento físico entre o estudante e o professor [Abreu-e Lima e Alves 2016]. O mesmo ocorre em avaliações cegas de trabalhos científicos, onde um parecer fornecido por um avaliador deve ser utilizado para verificar as

qualidades e defeitos do trabalho e propiciar as informações necessárias para a melhoria do mesmo. Normalmente, a avaliação de textos científicos (como projetos, artigos e relatórios) é feita avaliando-se um conjunto finito de tópicos (normalmente perguntas) que avaliam a qualidade da escrita, a estruturação utilizada ou a qualidade das referências citadas. Além disto, sempre há também campos para a escrita livre, onde o revisor (ou avaliador) justifica sua avaliação, indica sugestões e melhorias. Uma nota final é atribuída, ou de forma automática, em função das respostas dadas as perguntas, ou a partir de um valor atribuído pelo revisor. O texto dos campos “abertos” normalmente não são utilizados na atribuição da nota final da avaliação. Como o texto do revisor não passa por nenhum “processamento” ou verificação, podem ocorrer divergências entre a avaliação textual e a nota final obtida, a partir do conjunto finito de perguntas. Uma evolução interessante deste cenário, seria a avaliação automática do texto escrito pelo avaliador, não somente para verificar se este contém os elementos mínimos necessários à uma avaliação construtiva, mas também para confrontá-lo com a nota final atribuída ao texto. Neste artigo, apresentamos uma contribuição na busca da avaliação global dos textos escritos por avaliadores, utilizando para isso a mineração de opiniões.

A mineração de opiniões é definida por [Liu 2012] como a área que abrange todo estudo envolvendo uma análise computacional de opiniões, sentimentos, avaliações, atitudes e emoções. Essa área inclui diferentes aplicações, como sistemas de recomendação, marketing e análise de *feedback* de clientes [Cambria et al. 2013]. Alguns trabalhos também exploram a análise de *feedbacks* voltados a fins educacionais, como em [Rani e Kumar 2017] e [Dhanalakshmi et al. 2016].

Neste artigo propomos o desenvolvimento de um método utilizando mineração de opiniões para a avaliação automática de pareceres relacionados a trabalhos desenvolvidos por estudantes de iniciação científica escritos em português do Brasil. O método foi implementado em uma ferramenta capaz de avaliar um texto, indicando se este tem polaridade negativa ou positiva. Com o texto classificado, pode-se então verificar se a nota final atribuída “condiz” com a polaridade do texto e desta forma garantir que tanto a nota como o parecer estão condizentes entre si. Experimentos realizados com esta ferramenta mostraram que é possível classificar textos de caráter científico de modo que, ao gerar um modelo com um corpus de pareceres de projetos de iniciação científica, obteve-se uma precisão de 86% nas classificações de polaridade.

Este trabalho está organizado da seguinte forma: na Seção 2 são apresentados os trabalhos relacionados. A seção 3 apresenta um resumo sobre a tarefa de mineração de opiniões em textos. A seção 4 apresenta o método proposto neste trabalho. Na seção 5 são apresentados os experimentos realizados e os seus resultados. A última seção apresenta a conclusão e os trabalhos futuros.

2. Trabalhos Relacionados

Até o presente momento, não foi encontrado na literatura um trabalho utilizando a mineração de opiniões para classificação de pareceres de revisores de trabalhos científicos. Um trabalho correlato com a proposta apresentada aqui foi publicada por [Rani e Kumar 2017], onde um método para análise automática de *feedbacks* de estudantes é apresentado. O método analisa sentimento, emoção e satisfação em textos extraídos de questionários, fóruns de discussão, blogs e outras fontes.

Assim como Rani e Kumar outros autores exploraram a análise de textos escritos por estudantes. Em [Dhanalakshmi et al. 2016] a performance de diversos algoritmos de Mineração de Dados são comparadas para a tarefa de classificação de polaridade de comentários realizados por estudantes. No trabalho de [Dalal et al. 2015] é apresentado um método baseado em léxicos para a classificação de polaridade de *feedbacks* de cursos onlines escritos por estudantes.

Outra aplicação existente na área da educação é apresentada em [Tewari et al. 2015], onde os autores apresentam um sistema de recomendação com base nas opiniões dos alunos em relação ao material utilizado em aulas online. Além dos trabalhos apresentados outras formas de aplicação da Mineração de dados na área da educação são apresentadas em [Sachin e Vijay 2012]. Fora do contexto da educação existem diversas aplicações para a tarefa de mineração de opinião, como as apresentadas por [Pang et al. 2008].

Como visto em [Vinodhini e Chandrasekaran 2012], a maioria dos trabalhos envolvendo mineração de opiniões utiliza bases nos idiomas inglês e chinês. Embora esses idiomas sejam os mais utilizados diversos trabalhos da literatura apresentam o uso de outros idiomas, como o português. Em [Souza e Vieira 2012] os autores apresentam um estudo comparando diferentes recursos léxicos para classificação de polaridade em tweets escritos nesse idioma. O trabalho de [Freitas e Vieira 2013] também utilizou léxicos para português, mas aplicou o mesmo para a tarefa de classificação de polaridade de reviews de filmes. Uma abordagem utilizando aprendizagem de máquina foi apresentada por [Dosciatti et al. 2013], onde os autores apresentam o uso de SVM para classificar uma base de notícias rotulada com seis classes de emoções e polaridade.

3. A Mineração de Opiniões

O termo mineração de opiniões foi apresentado inicialmente por [Dave et al. 2003] e em muitos trabalhos possui o mesmo significado que o termo Análise de Sentimentos, introduzido por [Nasukawa e Yi 2003]. Com o passar do tempo, diversos outros termos foram criados para representar esse tipo de tarefa, como *opinion extraction*, *sentiment mining*, *subjectivity analysis*, *affect analysis*, *emotion analysis*, *review mining*, etc [Liu 2012].

Segundo [Liu 2012] essa área abrange todo estudo envolvendo uma análise computacional de opiniões, sentimentos, avaliações, atitudes e emoções relacionados a entidades. As entidades utilizadas para a análise podem ser produtos, serviços, organizações, indivíduos, questões, eventos, tópicos, etc. Devido a grande quantidade de objetos de estudo e aplicações, a utilização dos termos mineração de opiniões e análise de sentimentos acaba se tornando inadequada em alguns casos. No trabalho de [Munezero et al. 2014], os autores apresentaram definições específicas para cada um dos tipos de análises realizadas, possibilitando uma melhor diferenciação entre os trabalhos da área.

No contexto desta pesquisa, a mineração de opiniões é utilizada com o objetivo de identificar a polaridade de um texto. A polaridade positiva indica se um texto contém palavras relacionadas a emoções positivas como a *alegria*. Já a polaridade negativa indica que o texto contém palavras relacionadas a emoções negativas, tais como *medo*, *raiva*, ou *repugnância*.

Indiferente da terminologia utilizada, a mineração de opiniões possui alguns desafios que tornam essa tarefa complexa. Uma demonstração do nível de dificuldade exis-

tente pode ser visto no processo de anotação do corpus, onde diferentes avaliadores geralmente apresentam divergências ao realizar uma anotação em pares. A anotação em pares, pode ser entendida como uma forma de avaliação onde um texto é submetido para análise por vários avaliadores distintos.

A complexidade do processo se deve a falta de informações explícitas sobre as opiniões no texto. Em vários casos são apresentadas apenas informações subjetivas, necessitando de uma análise complexa para realizar a tarefa de classificação [Hu e Liu 2004]. Diversas vezes essas informações ainda sofrem influência de fenômenos como a ironia ou sarcasmo, alterando o significado do sentimento expresso (mudança da polaridade) e podendo gerar erros na classificação. Alguns trabalhos estudam a identificação desses casos [Carvalho et al. 2009] e a influência destes na tarefa de classificação [Maynard e Greenwood 2014]. Outro problema existente é a falta de estrutura dos textos. Muitos dos corpus encontrados na literatura, principalmente os extraídos de redes sociais, apresentam variações em relação ao tamanho e formas de escrita de cada texto. Em conjunto com os problemas citados anteriormente existem casos onde são utilizados gírias, abreviações e palavras incorretas, impedindo que todas as palavras sejam identificadas para a análise caso um tratamento não seja realizado, como em [Balahur e Turchi 2013]. Esses problemas podem ser ampliados em alguns casos onde o idioma nativo do usuário é diferente do idioma utilizado para escrever o texto, como apresentado por [Yim e Warschauer 2017].

Por fim, o fato das avaliações científicas possuírem uma linguagem mais técnica e formal, suaviza os problemas descritos acima. Entretanto, outros desafios acabam surgindo. O principal deles é justamente a sua linguagem mais “técnica”, que deve ser imparcial e ater-se somente a fatos científicos e conclusões lógicas. Isto torna complexo o processo de mineração de opiniões, uma vez que expressar emoções em textos científicos é mais raro.

Como apresentado em [Medhat et al. 2014], existem três abordagens principais para realizar a tarefa de mineração de opiniões: as baseadas em aprendizagem de máquina, as baseadas em léxicos e as abordagens que combinam as anteriores. A abordagem baseada em aprendizagem de máquina, utilizada nesta pesquisa, visa utilizar uma técnica de aprendizado de máquina para a tarefa de atribuição da polaridade de um texto. A vantagem dessa abordagem é não utilizar um léxico de palavras emocionais no processo, algo que nem sempre está disponível para a língua utilizada. Como apresentado no trabalho de [Aranha 2007], na maioria dos casos a abordagem baseada em aprendizagem de máquina é realizada em nível de documento, dividindo-se em etapas. A primeira etapa é referente a coleta da base de dados, que é seguida do pré processamento da mesma. Após o tratamento dos dados um algoritmo de mineração de dados é aplicado. Com a obtenção dos resultados é possível realizar a última etapa do processo, onde se analisam os resultados.

Na etapa de pré processamento diversas estratégias podem ser utilizadas para fornecer dados com maior qualidade e para aumentar a eficiência do processo de mineração subsequente. Como apresentado por [Ferreira et al. 2014], uma dessas estratégias é representar de forma genérica o máximo de palavras possíveis, diminuindo o número de palavras com poucas ocorrências no corpus. Uma técnica que permite essa forma de representação é a aplicação de um lematizador para extrair a raiz da palavra. Outra operação importante para possibilitar uma maior eficiência na tarefa de mineração de opiniões é a

remoção dos *stop words*, onde todas as palavras irrelevantes para a classificação são removidas. Assim como os *stop words*, as palavras menos frequentes também podem ser removidas. Outra forma de reduzir a quantidade de atributos é a utilização de um algoritmo para o cálculo do ganho de informação, que possibilita a seleção dos melhores atributos para a classificação.

Após o pré processamento um algoritmo de aprendizagem deve ser utilizado para realizar a classificação. Segundo [Medhat et al. 2014], os algoritmos mais comuns para esse tipo de tarefa são: Support Vector Machines (SVM), Naive Bayes (NB), K-Nearest Neighbour (KNN), Maximum Entropy (ME) e Redes Neurais. Neste trabalho optamos pelo uso do algoritmo SVM, utilizado com sucesso em pesquisas anteriores [Martinazzo et al. 2011]. De acordo com [Joachims 1998], o uso do SVM para a classificação de textos é uma escolha eficiente, tendendo a superar outros algoritmos de classificação. Segundo Joachims, algumas características do SVM tornam ele um algoritmo superior aos outros. O primeiro diferencial do SVM, é sua proteção contra overfitting, ou sobreajuste. O sobreajuste ocorre quando o modelo se ajusta excessivamente aos dados amostrais. Outro diferencial do SVM, é que através da contagem dos termos relevantes (ganho de informação), realizado no pré-processamento, o algoritmo é capaz de tratar quais termos são os termos mais ou menos relevantes na sua classificação. Por fim, Joachims afirma que conjuntos de textos são normalmente linearmente separáveis o que se torna uma ótima característica para o SVM.

4. Método Proposto

Nesta seção apresentamos o método desenvolvido para a avaliação de pareceres relacionados a trabalhos desenvolvidos por estudantes de iniciação científica escritos em português do Brasil. O método é baseado na aprendizagem de máquina supervisionada. Por isso, o método necessita de uma base de dados (chamada daqui em diante de corpus) para ser utilizada na construção de um modelo de predição.

4.1. Construção do corpus

Não há disponível na literatura um corpus com textos de avaliações de projetos científicos. Assim, a primeira etapa desta pesquisa consistiu na construção de um corpus com textos provenientes de avaliações de relatórios de projetos de iniciação científica obtidos junto ao Programa de Bolsas de Iniciação Científica da Pontifícia Universidade Católica do Paraná. Foram utilizados relatórios dos projetos no ano de 2015. Neste ano foram submetidos um total de 1.243 projetos distribuídos em áreas de conhecimento distintas. A Tabela 1 ilustra a distribuição das áreas de pesquisa. Os dados utilizados nesta pesquisa passaram por um processo de descaracterização quanto a identificação de sua autoria e área de conhecimento.

Para cada projeto, um par de avaliadores respondeu um questionário formado por três perguntas objetivas (múltipla escolha) e uma pergunta qualitativa (texto livre). Os revisores, na sua grande maioria professores dos programas de pós-graduação da PUCPR são orientados a, no campo de resposta livre, indicar os pontos positivos e negativos, analisando tecnicamente o relatório e justificando as escolhas feitas nas perguntas objetivas. Desta forma a base de dados deste trabalho pode ser dividida em duas partes: uma referente aos campos numéricos (escolhas das perguntas objetivas) e outra referente as respostas opinativas (texto em linguagem natural).

Tabela 1. Projetos Submetidos em 2015

Projetos Submetidos em 2015 (por Área de Conhecimento)			
	# de Submissões	# Aprovados	# Reprovados
Ciências Humanas	329	305	24
Ciências Exatas	179	169	10
Ciências Agrárias	124	119	5
Ciências da Vida	314	284	30
C.Sociais Aplicadas	297	287	10
Total	1243	1164	79

4.2. Pré processamento

O corpus original não passou por nenhum processo de rotulação ou anotação manual. Ou seja, dado um texto escrito por um revisor, não sabe-se a priori se este texto tem polaridade positiva ou negativa. Assim, um processo de rotulação automática precisa ser realizado antes que o corpus possa ser utilizado. Este processo consiste em analisar a nota individual de cada parecer, aplicando-se um rótulo ao parecer, que pode ser *aprovado* ou *reprovado*. O seguinte critério foi adotado para se aplicar os rótulos: projetos com notas maiores ou igual a 7.0 são classificados como aprovado e pareceres com nota inferior a 7.0 são rotulados como reprovado. Esses valores foram escolhidos por serem os mesmos utilizados pela instituição.

Após a rotulação do corpus, um pré-processamento foi realizado nos campos textuais de cada parecer avaliativo. A Figura 1 descreve as tarefas realizadas de forma sequencial nessa etapa.

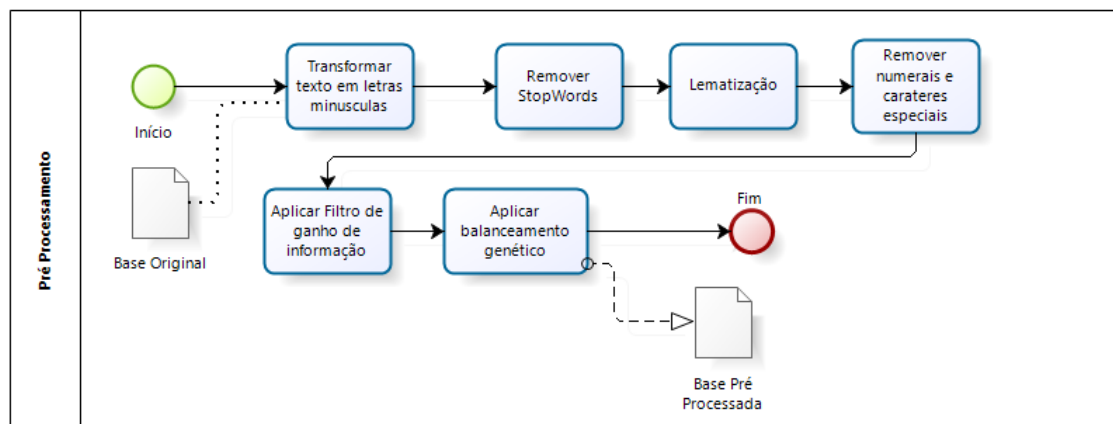


Figura 1. Etapas do pré-processamento dos textos

Para a remoção dos stop-words foi utilizada uma lista que contemplasse todas as palavras consideradas de baixa importância para o processo de classificação. Foi utilizada a lista disponível em SnowBall. Após a remoção das palavras irrelevantes para a classificação foi realizada a lematização das palavras restantes. Para esta tarefa foi utilizada a ferramenta SnowBall Stemmer. A lista de stop-words e a ferramenta de lematização estão disponibilizadas no site SnowBall¹. Por fim, todos caracteres que estavam em maiúsc-

¹<http://snowball.tartarus.org>

culo foram convertidos para minúsculo e todos os caracteres que possuíam acentos foram substituídos por sua versão sem acento. Para isto foi desenvolvida uma expressão regular.

4.3. Treinamento e teste

Após o pré processamento descrito na Seção 4.2, gerou-se um modelo de treinamento e teste empregando o algoritmo de classificação SMO (*Sequential Minimal Optimization*). O SMO é um algoritmo da família do SVM disponibilizado pelo Weka². Foram utilizados os parâmetros *default* do algoritmo. Para o treinamento e avaliação optou-se pela validação cruzada com 10 partições.

4.4. Construção da Ferramenta Web

Após a conclusão do pré-processamento e elaboração do modelo baseado no algoritmo SVM, iniciou-se o desenvolvimento de uma ferramenta web, para possibilitar a visualização dos resultados. Esta ferramenta possui duas funções principais. A primeira função é gerar um modelo para testes, utilizando uma base de dados no formato CSV, que é posteriormente convertida para formato ARFF. A segunda função da ferramenta, é realizar futuras classificações de pareceres elaborados pelos professores avaliadores.

Na sequencia, foram desenvolvidas interfaces para a utilização da ferramenta pelos professores. Nestas interfaces o professor avaliador possui campos disponíveis para responder as perguntas referentes ao projeto e desta forma atribuir uma nota e além disso também possui um campo textual com a função de preencher a sua opinião sobre o projeto em questão. Após o preenchimento das informações pelo usuário, estes dados são enviados, para um servidor onde é realizada a classificação do parecer recebido (projeto aprovado ou reprovado) e uma comparação do resultado da classificação com o a nota atribuída no projeto. Caso seja detectada alguma discrepância, como por exemplo, um projeto com uma nota elevada e uma classificação do projeto como reprovado, um alerta é emitido para que o usuário possa verificar se os campos preenchidos estão realmente consistentes. Caso nenhuma discrepância seja encontrada, o parecer e as notas são então registradas no banco de dados de projetos. Esta opção ainda não foi avaliada.

5. Resultados Experimentais

Neste artigo apresentamos os resultados de experimentos realizados com o objetivo de avaliar o processo de classificação dos textos (identificação da polaridade). Não avaliamos ainda a utilização desta ferramenta pelos professores avaliadores de projetos de iniciação científica.

Inicialmente é interessante constatar um detalhe sobre os textos dos pareceres: a média de palavras utilizadas nos textos escritos pelos avaliadores varia se o relatório é aprovado ou não. Enquanto nos relatórios aprovados obteve-se uma média no comprimento dos textos de 39 palavras, nos projetos reprovados a média foi de 99 palavras. Uma possível justificativa para isto, é o fato de um professor revisor ao reprovar um relatório, embasar com mais detalhes a sua opinião de modo a justificar o porque de ter avaliado negativamente um trabalho. Em trabalhos futuros esta característica será melhor estudada. Além disto, estes valores indicam que a ferramenta desenvolvida pode “sugerir”

²<http://www.cs.waikato.ac.nz/ml/weka/>

Tabela 2. Resultados Obtidos com SVM

	Precisão	Cobertura	Medida F
Aprovado	0,917	0,929	0,923
Reprovado	0,489	0,466	0,466
Média	0,86	0,865	0,862

Tabela 3. Matriz de Confusão

	Aprovado	Reprovado
Aprovado	902	69
Reprovado	82	66

que o comprimento mínimo de um parecer não deveria ser menor que aproximadamente 40 palavras.

O corpus utilizado no experimento tem 1.119 textos (revisões de projetos). Alguns dos 1.243 projetos submetidos foram iniciados e não tiveram sua continuidade, levando a não avaliação de um relatório sobre o mesmo. A tabela 2 apresenta os resultados do experimento que avalia o modelo criado a partir do corpus de avaliações. Duas métricas são importantes de serem analisadas. A primeira é a precisão obtida que foi de 86,0% e a segunda o valor da Medida-F que foi de 86,2%. Os resultados podem ser considerados satisfatórios para o projeto proposto, entretanto algumas considerações devem ser feitas. Como é possível observar na Tabela 3, a base de dados utilizada é significativamente desbalanceada, de modo que o classificador está tendendo a classificar os pareceres dos avaliadores como aprovado na maioria das vezes. Isto se comprova ao analisar a precisão obtida nas classificações das instancias reprovadas que foi de 48,9%.

Uma análise inicial dos textos avaliados como negativos mostra a presença de um conjunto mais frequente de palavras: “relatório”, “resultados”, “resumo” e “discussão”. Em conjunto com estas palavras, observou-se que elas são seguidas de palavras negativas como por exemplo “erros”, “ausência”, “refazer” e principalmente a negação “não”. Desta forma observa-se que as críticas apontadas pelos avaliadores, são frequentemente direcionadas a seções inteiras dos relatórios (introdução, discussão, resultados...). Em contrapartida, relatórios aprovados normalmente também possuem alguma crítica ou sugestão de melhoria do projeto e/ou relatório. Entretanto, estas por sua vez, são direcionadas para problemas pontuais do projeto e/ou relatório. A Tabela 4 apresenta trechos de avaliações de relatórios. O exemplo 1 apresenta uma avaliação de projeto aprovado e que foi corretamente classificada como positiva pelo método. O exemplo 2 também foi corretamente classificado como texto negativo. O exemplo 3 apresenta uma avaliação negativa por parte do revisor, mas que foi classificada como positiva pelo método. Já o exemplo 4 apresenta um projeto aprovado pelo revisor e que recebeu rótulo negativo pelo método.

6. Conclusão

Este artigo apresentou um estudo sobre a utilização da mineração de opiniões na avaliação de pareceres de projetos de iniciação científica. Trata-se de um estudo inicial com vistas a melhoria do processo de avaliação de pareceres científicos, uma vez que discrepâncias podem ocorrer entre os textos dos revisores e a nota atribuída ao projeto. Apesar da

Tabela 4. Exemplos de textos do Corpus

Exemplo 1	O trabalho como um todo atingiu os objetivos propostos e foi finalizado satisfatoriamente. Sugere-se para a compreensão mais ampla e para finalidade de uma acuidade científica maior, detalhar principalmente o material e método...
Exemplo 2	O resumo apresentado não apresenta as informações básicas necessárias. O método não foi descrito em profundidade e não é possível saber quantos telejornais foram assistidos para fazer a análise (a amostra não foi estabelecida/calculada). Os instrumentos de pesquisa a serem utilizados para a análise do material não foram apresentados, o processo de análise não foi claramente descrito ...
Exemplo 3	A não pesagem, conforme proposto no projeto, durante todo o período de 2014 a 2015, inviabiliza a realização de análise estatística e, conseqüentemente, da interpretação dos dados. O trabalho não possui dados suficientes para análise estatística.
Exemplo 4	O relatório final está bem escrito e bem organizado. No entanto, a discussão está longa em virtude do número de resultados, o que dificulta a identificação dos principais resultados do trabalho. Os resultados das análises estatísticas não estão apresentados no artigo, apesar de estar descrito que foram realizadas. Sugiro a inclusão dos mesmos para a apresentação na...

ferramenta ainda não ter sido aplicada em um processo real de avaliação, dados reais de avaliações passadas (2015) mostraram a viabilidade do projeto.

As próximas etapas deste projeto apresentam a avaliação do mesmo durante um processo de avaliação de relatórios de projetos de iniciação científica. Também deve-se explorar a utilização de outros algoritmos para avaliar o modelo de predição, como por exemplo o random forests e redes neurais.

Outro estudo que já foi iniciado e que será objeto de publicação futura diz respeito a uma análise mais específica sobre as palavras e expressões utilizadas nos pareceres que foram aprovados ou reprovados. Este estudo visa ajustar a ferramenta que avalia o texto dos avaliadores de modo a, por exemplo, indicar a presença de palavras inapropriadas nos textos das avaliações. Muitas vezes, mesmo sem perceber, um avaliador pode utilizar palavras muito “negativas” o que pode incomodar o leitor (por exemplo o aluno que recebe a avaliação do seu relatório) mais sensível. Para tal, estamos avaliando a presença das emoções em cada texto. As emoções podem dar mais subsídios sobre o quão negativo é um texto, uma vez que elas podem indicar palavras associadas a emoção *raiva* ou *tristeza*, por exemplo.

Referências

- Abreu-e Lima, D. M. and Alves, M. N. (2016). O feedback e sua importância no processo de tutoria a distância. *Pro-Posições*, 22(2):189–205.
- Aranha, C. N. (2007). *Uma abordagem de pré-processamento automático para mineração de textos em português: sob o enfoque da inteligência computacional*. 2007. 144 f. PhD thesis, Tese (Doutorado em Engenharia Elétrica)–Programa de Pós-Graduação em Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.
- Balahur, A. and Turchi, M. (2013). Improving sentiment analysis in twitter using multi-lingual machine translated data. In *RANLP*, pages 49–55.
- Cambria, E., Schuller, B., Xia, Y., and Havasi, C. (2013). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2):15–21.
- Carvalho, P., Sarmiento, L., Silva, M. J., and De Oliveira, E. (2009). Clues for detecting irony in user-generated contents: oh...!! it’s so easy;- . In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM.
- Dalal, R., Safhath, I., Piryani, R., Kappara, D. R., and Singh, V. K. (2015). A lexicon pooled machine learning classifier for opinion mining from course feedbacks. In *Advances in Intelligent Informatics*, pages 419–428. Springer.
- Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM.
- Dhanalakshmi, V., Bino, D., and Saravanan, A. M. (2016). Opinion mining from student feedback data using supervised learning algorithms. In *3rd MEC International Conference on Big Data and Smart City*, pages 1–5. IEEE.

- Dosciatti, M. M., Ferreira, L. P. C., and Paraiso, E. C. (2013). Identificando emoções em textos em português do brasil usando máquina de vetores de suporte em solução multiclasse. *ENIAC-Encontro Nacional de Inteligência Artificial e Computacional. Fortaleza, Brasil.*
- Ferreira, L. P. C., Dosciatti, M. M., and Paraiso, E. C. (2014). Estudo do impacto de um corpus desbalanceado na identificação de emoções em textos. In *Encontro Nacional de Inteligência Artificial e Computacional*, pages 394–400. SBC.
- Freitas, L. A. and Vieira, R. (2013). Ontology based feature level opinion mining for portuguese reviews. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 367–370. ACM.
- Hu, M. and Liu, B. (2004). Mining opinion features in customer reviews. In *AAAI*, volume 4, pages 755–760.
- Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine learning*, pages 137–142.
- Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.
- Marques, J. P., Del’Arco, L., Martini, M. L., PAIVA, M. L. M. F., and De Rose, T. M. S. (2006). Intervenção para a promoção do autoconceito: combinando feedback focalizado internamente e feedback atribucional em situação de tutoria de leitura. In *Associação Brasileira de Psicologia Escolar e Educacional*.
- Martinazzo, B., Dosciatti, M. M., and Paraiso, E. C. (2011). Identifying emotions in short texts for brazilian portuguese. In *IV International Workshop on Web and Text Intelligence*.
- Maynard, D. and Greenwood, M. A. (2014). Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *LREC*, pages 4238–4243.
- Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113.
- Munero, M. D., Montero, C. S., Sutinen, E., and Pajunen, J. (2014). Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE transactions on affective computing*, 5(2):101–111.
- Nasukawa, T. and Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM.
- Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135.
- Perrenoud, P., Thurler, M. G., de Macedo, L., Machado, N. J., and Alessandrini, C. D. (2009). *As competências para ensinar no século XXI*. Artmed Editora.
- Rani, S. and Kumar, P. (2017). A sentiment analysis system to improve teaching and learning. *Computer*, 50(5):36–43.

- Sachin, R. B. and Vijay, M. S. (2012). A survey and future vision of data mining in educational field. In *Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on*, pages 96–100. IEEE.
- Souza, M. and Vieira, R. (2012). Sentiment analysis on twitter data for portuguese language. *Computational Processing of the Portuguese Language*, pages 241–247.
- Tewari, A. S., Saroj, A., and Barman, A. G. (2015). e-learning recommender system for teachers using opinion mining. In *Information Science and Applications*, pages 1021–1029. Springer.
- Vinodhini, G. and Chandrasekaran, R. (2012). Sentiment analysis and opinion mining: a survey. *International Journal*, 2(6):282–292.
- Yim, S. and Warschauer, M. (2017). Web-based collaborative writing in l2 contexts: Methodological insights from text mining. *Language Learning Technology*, 21(1):146—165.

Improved User Cold-Start Recommendation via Two-Level Bandit Algorithms

Otávio Augusto Rodrigues¹, Anísio Lacerda¹, Flávio Luis Cardeal Pádua¹

¹Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas, 7675 – Nova Gameleira – Belo Horizonte – MG – Brazil

otavio.augusto@outlook.com, {anísio, cardeal}@decom.cefetmg.br

Abstract. *Recommender systems have been gaining visibility over the years due to its ability in helping users to deal with the information overload problem. However, the recommendation to new users (user cold-start) is still an open problem for these systems. The user cold-start problem is recognized as an exploration/exploitation dilemma, for which the system needs to balance (i) maximizing the user satisfaction (exploitation) and (ii) learn about users' tastes (exploration). We propose a new bandit algorithm that suggests items to users as a two-step process. First, we select the most relevant cluster to the target user, then, given the selected cluster, we choose an item from it that matches the user's tastes. The experimental evaluation shows that our strategy yields significant improvements regarding recommendation quality over the state-of-the-art bandit algorithms.*

1. Introduction

The aiming of recommender systems is to recognize relevant items according to user's tastes. Recommender systems are based on users' explicit and implicit tastes, the preferences of other users, and user and item attributes. For instance, a movie recommender system can incorporate both explicit rating data (e.g., Alice rates *Toy Story* a 4 out of 5), and movie content information (e.g., *Forrest Gump* is targeted as comedy) to suggest movies that match users' tastes.

There is inherent uncertainty in the user's preference information, which can be gathered through online interaction with users and real-time adaptation of recommendation models. Furthermore, a considerable number of users might be completely new to the recommender system, i.e., there is insufficient information about consumption history. This scenario is known as the user *cold-start* problem [Schein et al. 2002] and represents a challenging problem for recommender systems. In a user *cold-start* setting, traditional recommendation approaches suffers from learning a matching function for users' tastes and items popularity.

In Figure 1, we present the distribution of ratings per users for the MovieLens1M dataset [Miller et al. 2003]. As we can see, few users rates (i.e., evaluates) a large number of movies; whereas, a lot of users rates a small number of movies. This distribution is an example of the user cold-start problem, for which there is severe scarcity in the preferences from users to items.

The user cold-start scenario represents a challenge to recommender systems, and are recognized as an exploration/exploitation problem. This problem refers to find a trade-

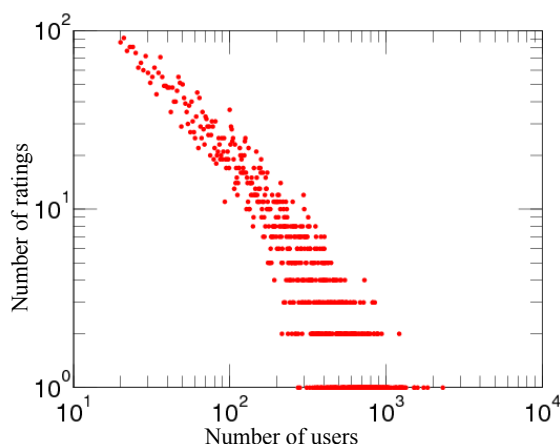


Figure 1. Distribution of ratings per users.

off between two competing goals: (i) maximizing users satisfaction, in the long run, while (ii) exploring uncertainties of user tastes [Agarwal et al. 2009]. For instance, a movie recommender system should find the most relevant movie to users while keep trying to improve the understanding of their preferences.

Traditionally, the exploration/exploitation dilemma is formulated as a multi-armed problem [Li et al. 2010, Tang et al. 2013, Vermorel and Mohri 2005]. In our context, i.e., recommender systems, we state the problem as: for each recommendation round, the algorithm selects an item (e.g., a movie) to pull (i.e., recommend), and after pulling, it receives a reward. Hence, personalized recommendation can be seen as an instantiation of the multi-armed bandit problem. Note that the reward is drawn from an unknown probability distribution determined by the selected item, and it refers to the user response (e.g., a click). The reward is fed back to the bandit algorithm and used to optimize its recommendation strategy. Shortly, the optimal strategy is to pull the arm with the maximum expected reward with respect on the user on each round, and thus maximize the total cumulative reward for the set of rounds.

Typical solutions of the multi-armed bandit problem assume that the arms are independent [Auer et al. 2002]. In this work, we drop this assumption and focus on the bandit problem for which the arms are *dependent*. This dependent-arm assumption has practical scenarios of interest. For instance, we can cast the news recommendation problem as a bandit problem where each news article refers to an arm, showing an article corresponds to pulling an arm, and user clicks are the reward. News articles with similar textual content are likely to receive similar amount of clicks, and this creates dependencies between the arms.

In this paper, to capture the dependence among items, we propose a generalized two-level clustering-based bandit framework. More specifically, first, we consider the arms as clusters of items and use a bandit algorithm to select a group of items to be recommended. Hence, given a pulled cluster, we use a distinct bandit algorithm to select the most relevant item within this cluster. Our proposed framework is flexible to consider, as input, (i) a clustering method and (ii) two bandit algorithms (i.e., one for each level), which leads to several algorithms as instantiations of the framework.

In summary, the contribution of this work is twofold:

- We propose a new bandit algorithm to deal with user cold-start, which is based on a two-level recommendation decision process of clustered items.
- We conduct extensive experiments on a real-world movie dataset to validate the accuracy of the proposed family of algorithms.

The paper is organized as follows. In Section 2, we present the related previous work. In Section 3, we detail our proposed bandit algorithm for the recommendation in user cold-start scenario. In Section 4, we present the experimental setup. In Section 5, we list and discuss the achieved results. Finally, in Section 6, we conclude the paper and present possible future lines of research.

2. Related work

In the following, we highlight the previous research that is most relevant to our work.

Bandits for recommendations: In the recommender systems literature, there is an extensive literature of methods conceived to perform online recommendations. As already mentioned, the most common approach to deal with online recommendation is modeling the task as a multi-armed bandit problem [Bouneffouf et al. 2012, Li et al. 2010]. In [Li et al. 2010], the authors first introduced the online new recommendation problem using a bandit formulation, which first mention the contextual multi-armed bandit algorithms. In this case, the context refers to information about users. We emphasize that our approach is *context-free*, i.e., we assume that all users are new, for which there is no historic information. Hence, we are unable to compare our approach to contextual multi-armed bandit algorithms, since the later assume that there is historic information about users.

User cold-start problem: The user cold-start problem is one of the most challenge problems in recommender systems. Hence, there have been several approaches proposed to deal with this scenario of scarcity of information [Park et al. 2006, Leroy et al. 2010]. For instance, in [Park et al. 2006], the authors proposed the usage of filterbots, which are bots that automatically inject ratings into the recommender system to reduce data sparsity. Hence, this simple approach inserts “fake” ratings into the system to improve the quality of cold-start recommendations. In a different scenario, i.e. friend suggestion in social networks, the authors in [Leroy et al. 2010], investigated the cold-start problem using group membership preferences among peers. In this work, different from previous approaches, we focus on online recommendation.

3. TL-Bandits

In this section, we formalize the problem investigated in this paper and present relevant concepts critical to understanding the on-line recommendation task.

3.1. Problem Formalization

Formally, let U be an arbitrary input space and $I = \{1, \dots, M\}$ be a set of M arms, where each arm corresponds to an item. Let E be a distribution over tuples (u, r) , where $u \in U$ corresponds to the target user, i.e., the user we are interested in recommending items to,

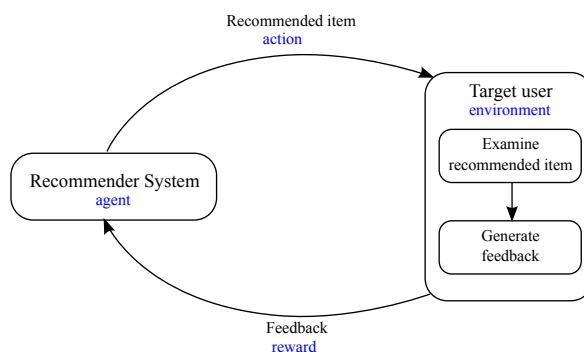


Figure 2. The online recommendation task modeled as a Multi-armed Bandit problem.

and $r \in \{0, 1\}$ refers to the reward received by the algorithm. Hence, the reward is 1 if the user considers that the suggested item is relevant, and 0 otherwise.

In Figure 2, we present the interaction cycle between a target user and an on-line recommender system. The user arrives, and the recommender system selects a relevant item, which is presented to the user. The user interacts with the list by clicking on the suggested item whether it is relevant to him. The interaction is interpreted as feedback about the quality of the recommended item. Hence, this feedback is used to update the recommendation model aiming to provide better recommendations in the future. The cycle finishes and the next user is considered. This formulation translates to the Reinforcement Learning problem (see the terminology in blue color in Figure 2).

3.2. Multi-armed Bandits

The online recommendation task is often modeled as a Multi-armed Bandit problem, which originates from the reinforcement learning paradigm. The MAB problem is stated as a sequential Markov Decision Process (MDP) of an agent that tries to optimize its actions while improving its knowledge on the arms. The central challenge in bandit problems is the need for balancing exploration and exploitation. Interactively, the algorithm \mathcal{A} *exploits* past knowledge to choose the arm that seems best. On contrary, this apparently optimal arm may be suboptimal choice, due to imprecisions in \mathcal{A} 's knowledge. To avoid this undesired situation, \mathcal{A} has to *explore* by selecting seemingly suboptimal arms to gather more information about them.

In Algorithm 1, we present the generalized MAB algorithm. A MAB algorithm iterates in discrete rounds $t = 1, 2, \dots$ (Line 1). In each round t , the algorithm selects an item $i(t) \in I$ based on target user $u(t)$ and on the knowledge, it collected from the previous rounds (Line 2) and presents the selected item to the target user (Line 3). Hence, the reward $r(t)$ ¹ is revealed (Line 4–8). Finally, the algorithm improves its arm-selection policy with the new observation $u_t, i(t), r(t)$ (Line 9). Note that the algorithm chooses a single item per round.

3.3. Two-Level Bandit Algorithms (TL-Bandits)

In Figure 3, we present the overall scheme of our proposed Two-Level Bandit algorithm. As we can see, the selection of which item suggest to the target user is performed in two

¹We use $r(t)$ as the reward for item $i(t)$ in trial t

Algorithm 1 Generalized Multi-armed Bandits**Require:** Bandit algorithm \mathcal{A}

```

1: for  $t = 1, \dots, T$  do
2:    $i(t) \leftarrow \text{select-item}(u(t), \mathcal{A})$ 
3:   show  $i(t)$  to user; register click
4:   if user clicked  $i(t)$  then
5:      $r(t) = 1$ 
6:   else
7:      $r(t) = 0$ 
8:   end if
9:   update-mab( $u(t), i(t), r(t), \mathcal{A}$ )
10: end for

```

steps. First, we need to select the cluster that is most likely to contain a relevant video to the user. Hence, given the selected cluster, we have another step that is the selection of a video within the chosen cluster. In both levels, we have a MAB algorithm that is responsible for selecting a cluster (level 1) or a video (level 2). However, the reward is dependent on the choice made by the MABs.

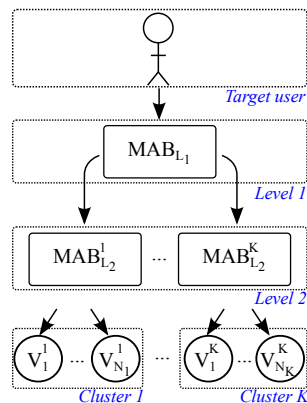


Figure 3. TL-Bandits overview. We assume K clusters of items.

In Algorithm 2, we detail the online recommendation problem as a Two-Level bandit algorithm. Note that our algorithm assumes as input two bandit algorithms, one for each level. We also consider that the items (i.e., videos) were previously clustered. Hence, as in Algorithm 1, the recommendation process iterates in discrete rounds $t = 1, 2, \dots$ (Line 1). In each recommendation round t , the algorithm first selects a cluster from which an item will be selected (Line 2). In Line 3, the MAB from level 2 then selects a video within the previously selected cluster. The selected video is presented to the target user, and their feedback is captured (Lines 4–9). The final step refers to updating the knowledge for the MABs in each level independently (Lines 10–11). In this work, we assume that the MABs in both levels are from the same family, i.e., if we have the Bayes UCB algorithm in level 1, then we have another instantiation of the same algorithm in level 2. We emphasize that the proposed TL-Bandits are flexible to consider any MAB algorithm to choose both the cluster and the videos within the clusters.

Algorithm 2 Two-Level Bandit Algorithm

Require: Bandit algorithm level-1 \mathcal{A}_1 and level-2 \mathcal{A}_2

```

1: for  $t = 1, \dots, T$  do
2:    $c(t) \leftarrow \text{select-cluster}(u(t), \mathcal{A}_1)$ 
3:    $i(t) \leftarrow \text{select-item}(u(t), c(t), \mathcal{A}_2)$ 
4:   show  $i(t)$  to user; register click
5:   if user clicked  $i(t)$  then
6:      $r(t) = 1$ 
7:   else
8:      $r(t) = 0$ 
9:   end if
10:  update-mab( $u(t), i(t), r(t), \mathcal{A}_1$ )
11:  update-mab( $u(t), i(t), r(t), \mathcal{A}_2$ )
12: end for

```

4. Experimental setup

In this section, we describe our experimental setup.

4.1. Research Questions

The research questions guiding the remainder of the paper are:

- RQ1** Do clustering approaches improve the state-of-the-art of online recommendation methods? (See Section 5.1)
- RQ2** What is the effect on recommendation accuracy of varying the clustering method? (See Section 5.2)
- RQ3** What is the effect on recommendation accuracy of varying the number of clusterings? (See Section 5.3)
- RQ4** What is the effect on recommendation accuracy over the rounds? (See Section 5.4)

4.2. Dataset

To assess the recommendation quality of our method, we used the MovieLens1M dataset, which contains 1,000,209 ratings of 3,883 movies made by 6,040 users [Miller et al. 2003]. The collected ratings are in a 1-to-5 star scale. We emphasize that this is a publicly available movie rating dataset.

4.3. Metrics

In the context of item recommendation, when an item is clicked, a reward of 1 is incurred; otherwise the reward is 0. Hence, the expected reward of an item is its *Click-Through Rate* (CTR), and choosing an item with maximum predicted CTR is the same as to maximizing the expected number of clicks from users, which in turn equivalent to maximizing the total expected reward of our bandit formulation.

The averaged reward is equivalent to the metric CTR, which is the total reward divided by the total number of trials:

$$\text{Relative CTR} = \frac{1}{T} \sum_{t=1}^T r_t, \quad (1)$$

where r_t is the reward at round t , and T is the total number of rounds. We report the algorithm *relative CTR*, which is the algorithm's CTR divided by the random recommender.

We also report results based on the *Cumulative Click-Through Rate*, which refers to the sum of rewards received by a MAB algorithm until round i . Formally:

$$\text{Cumulative CTR}(i) = \sum_{t=1}^i r_t, \quad (2)$$

In other words, the Cumulative CTR measures the total amount of reward that an algorithm receives up until some fixed point in time.

4.4. Multi-armed Bandit Algorithms

Following, we detail the context-free MAB algorithms used to instantiate our framework.

- **Random**: it randomly chooses an item to pull.
- **ϵ -greedy (ϵ) (EG)**: it chooses a random item with probability ϵ , and chooses the arm with the highest CTR estimate with probability $(1 - \epsilon)$ [Tokic 2010].
- **UCB1**: it chooses the item with the highest UCB (*Upper Confidence Value*) score, according to [Auer et al. 2002].
- **UCB2 (η)**: the same as the above, but with a different UCB score computation [Auer et al. 2002]. The η parameter balances the amount of exploration and exploitation.
- **Thompson sampling (TS)**: it is a bayesian implementation of the MAB algorithm in which the (unknown) reward values are inferred from past data and summarized using a posterior distribution. It selects the item proportionally to its probability of being optimal under this posterior [Thompson 1933].
- **Bayes UCB**: it is a Bayesian adaptation of the UCB algorithm, in which the upper confidence bound score is computed by taking into account an upper quantile on the posterior mean [Kaufmann et al. 2012].

4.5. Clustering Methods

To instantiate our framework, we use three distinct clustering strategies, which are detailed in the following.

4.5.1. Category

In this clustering approach, we use the pre-defined category in which all movies in the dataset are classified. For instance, we have the categories *Action*, *Adventure*, *Romance*, etc. Please, refer to [Miller et al. 2003] to see the complete list of categories. The main disadvantages of this clustering approach are that (i) the number of clusters is fixed, and (ii) it is costly, since we need a manual classification for each item.

4.5.2. k-Means

Let $M = \{m_1, \dots, m_n\}$ be the set of movie title and synopses concatenated for each movie m_i , for which we want to compute a set of k clusters, $C = c_1, \dots, c_k$ [Jain 2010].

The k-Means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points within the cluster is minimized. Given μ_k be the centroid of cluster c_k , the squared error $E(c_k)$ between μ_k and the points in cluster c_k is defined as

$$E(c_k) = \sum_{m_i \in c_k} \|m_i - \mu_k\|^2. \quad (3)$$

The final goal of k-Means is to minimize the sum of the squared error over all K clusters,

$$E(C) = \sum_{k=1}^K \sum_{m_i \in c_k} \|m_i - \mu_k\|^2. \quad (4)$$

4.5.3. Latent Dirichlet Allocation (LDA)

The Latent Dirichlet Allocation topic model is a generative probabilistic model for collections of discrete data (e.g., text corpora) [Blei et al. 2003]. A topic model is an algorithm that focuses on discovering latent structures, i.e., hidden patterns that help to describe the data. Since it is a generative model, it defines how words in a document are generated through the control of latent topics. In the context of movie recommendation, the words come from movie title and synopsis, and the topics refer to groups of movies with similar characteristics.

In LDA, it is assumed that each observed movie description (title and synopsis) was generated by weighted mixtures of unobserved latent topics, which can be learned from the documents and refers to semantic themes present in the movie dataset. Formally, the LDA is defined using the following terms [Blei et al. 2003]:

- A *word* is the basic unit of discrete data from a vocabulary V .
- A *movie description* is a sequence of N words extracted from movie's title and synopsis.
- A *dataset* is a collection of $|I|$ movie descriptions.

The generative process of LDA specifies a simple probabilistic procedure to generate new documents according to a set of topics ϕ . The movie description are generated by first selecting a distribution over topics θ from a Dirichlet distribution, which determines the probability $P(z)$ (probability of topic z) for words in that description. The words in the movie description are then generated by selecting a topic i from this distribution and then picking a word from that topic following $P(w|z = i)$. The estimation of model parameters used here follows the Collapsed Gibbs Sampling approach [Griffiths and Steyvers 2004].

4.6. Evaluation Protocol

The evaluation of online learning algorithms with past data is a challenging task. Here, we use the state-of-the-art method for offline evaluation of multi-armed bandit algorithms, namely, *BRED (Bootstrapped Replay on Expanded Data)* method. The BRED method is based on bootstrapping techniques and enables an unbiased evaluation by utilizing historical data. The method assumes that the dataset is static to perform the bootstrap resamples.

Table 1. Average Relative CTR. In bold we highlight the gains over the baseline. In parentheses we show the number of clusters.

	Baseline -	Category (17)	k-Means Clustering						LDA Clustering					
			(5)	(10)	(17)	(25)	(50)	(100)	(5)	(10)	(17)	(25)	(50)	(100)
Random	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
EG(0,0)	1,146	3,752	1,326	0,233	2,974	0,323	0,364	0,010	2,126	0,045	0,045	0,138	0,000	0,087
EG(0,1)	1,034	6,296	1,670	1,757	3,466	4,836	4,248	3,182	6,551	5,568	6,348	5,601	5,026	6,317
EG(0,2)	0,921	5,872	1,545	1,929	3,345	4,079	3,467	4,924	5,669	5,288	6,536	4,978	5,517	6,260
EG(0,3)	0,966	4,576	1,602	1,981	2,569	3,360	3,486	4,667	4,795	3,909	4,446	4,290	4,810	5,308
EG(0,4)	0,989	3,608	1,511	1,871	2,586	3,503	3,168	2,444	3,819	3,432	4,259	3,043	3,698	3,817
EG(0,5)	0,933	3,248	1,371	1,743	1,629	3,053	2,364	3,298	3,472	2,750	2,205	2,036	2,879	3,490
EG(1,0)	0,944	1,008	0,947	0,929	1,009	1,116	0,841	0,803	0,992	0,977	1,036	1,014	0,931	0,904
UCB1	0,876	1,824	0,852	0,833	1,560	0,910	0,864	0,955	2,835	1,818	1,411	1,275	1,328	1,452
UCB2 (0,0)	2,135	3,752	1,326	0,233	2,974	0,323	0,364	0,010	2,126	0,045	0,045	0,138	1,681	0,087
UCB2 (0,1)	1,011	2,104	1,061	1,033	1,690	1,048	1,028	1,020	3,630	2,220	1,679	1,522	1,422	1,798
UCB2 (0,2)	0,831	2,040	1,000	1,057	1,707	1,063	1,061	1,081	3,748	2,083	1,634	1,449	1,543	1,538
UCB2 (0,3)	0,978	2,024	1,027	1,090	1,595	1,111	1,028	0,980	3,567	2,167	1,563	1,551	1,491	1,462
UCB2 (0,4)	0,933	2,216	1,030	0,976	1,750	1,201	0,972	0,975	3,567	2,000	1,598	1,725	1,543	1,615
UCB2 (0,5)	1,135	2,112	1,057	0,981	1,931	1,106	0,986	1,015	3,709	1,977	1,813	1,406	1,474	1,846
UCB2 (0,7)	1,079	2,216	0,932	1,124	1,948	1,196	1,056	1,020	3,614	1,932	1,598	1,565	1,672	1,769
UCB2 (1,0)	1,045	2,192	1,102	1,167	1,983	1,317	1,276	1,172	3,386	2,106	1,688	1,609	1,741	1,904
Bayes UCB	0,697	6,344	1,652	1,862	2,897	3,804	2,430	2,157	7,551	6,811	7,107	5,181	4,267	2,846
TS	1,079	1,272	1,856	2,205	1,207	4,529	2,907	3,616	1,016	0,970	1,098	1,246	1,181	1,231

We follow the same approach presented in [Mary et al. 2014], and assume the static world on small portions of the MovieLens1M dataset. We took the smallest number of portions such that a given portion has a fixed number of movies. The BRED evaluation method stems from the idea of bootstrapping, first introduced by [Efron 1979]. Let us consider each portion as a single dataset D of size T with K possible choices (arms). Then, from D , at each step, generate P new sampled datasets D_1, D_2, \dots, D_P of size $K \times T$ by sampling with replacement. For each sampled dataset i of size D_i with K_i items, we compute the estimated click through-rate (CTR) by averaging the CTR of each bandit algorithm on 100 random permutations of the data. We report the average CTR and cumulative CTR over all portions.

5. Results and Discussion

5.1. Performance of baseline recommendation methods

In Table 1, we present the CTR performance metric of all tested bandit algorithms. Note that, all results are normalized about the Random method. Hence, strategies with numbers below 1.0 are worst, regarding CTR, than the Random method. The Baseline column refers to the bandit strategies without considering clusters of items, i.e., they refer to strategies for which the arm is the item. In bold we highlight the results for which we achieve a better result than the baselines.

We first consider the clusterings as the category of the movies (e.g., Drama, Comedy, ...). When using the category to clusters the movies, we have a pre-defined number of groups. In our dataset, the movies are grouped into 17 categories. When using this instantiation of our framework, we achieve promising results regarding CTR. For instance, the Bayes UCB algorithm achieves an average relative CTR of 6,344.

We also tested two other clustering approaches for which we can define the number of clusters, i.e., k-Means and LDA. When analyzing the best results, we can see that

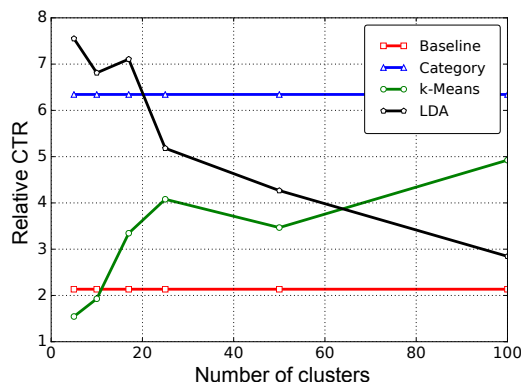


Figure 4. CTR vs Number of Clusters. The MAB algorithm used with Baseline, Category, k-Means, and LDA, are UCB2(0.0), Bayes UCB, EG(0.2), and Bayes UCB, respectively.

the LDA with 5 clusters outperforms all tested configurations of our proposed framework and all instances of the baseline.

5.2. Effect of the clustering methods

As presented in Table 1, the best results regarding CTR refers to clustering items with LDA topic representation. Remembering that LDA extracts topics from movie synopses and represent them in a latent space. We use this representation to describe the movies and cluster them according to the most probable topic. An advantage of LDA and k-Means is that the number of clusters is flexible and can be tuned according to each dataset. We attribute the best results of LDA to its robustness in dealing with the highly sparse description of movies' synopses.

We can also see from Table 1 that the performance of k-Means is better than the baseline. However, when compared with both Category and LDA clustering strategies, the k-Means presents was unable to present better recommendations. We believe that the sparsity of textual representation of items leads to the poor items grouping generated by the k-Means algorithm.

5.3. Effect of the number of clusters

In Figure 4, we present the relative CTR values regarding the number of clusters. Note that both the baselines and the category-based clustering approach are independent of the number clusters. In the former, i.e. the baselines, each item is an arm, hence there are no clusters. In the latter, i.e. the category-based approach, the number of clusters is pre-defined and equal to 17 in our dataset. Hence, for these cluster insensitive approaches, in Figure 4, we have two lines parallel to the x-axis.

When considering the k-means and LDA clustering approaches, we have different patterns. Whereas in k-Means, as we increase the number of clusters, we see improvements in performance; in the LDA, we have the opposite trend, i.e., the best performance of LDA happens when we have only 5 clusters.

5.4. Analysis of recommendation quality through rounds

In this section, we investigate the behavior of each recommendation strategy as the number of rounds evolves. From Figure 5, note that each round refers to a specific interaction

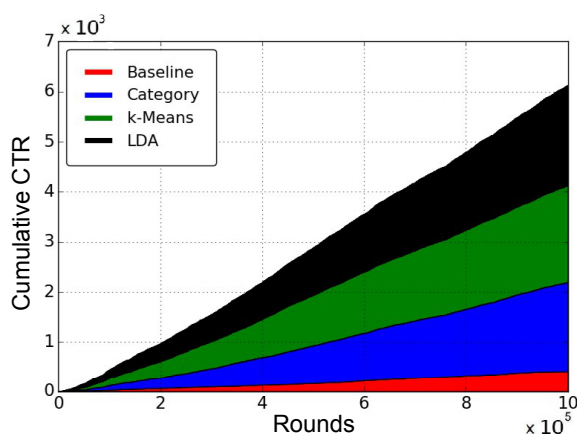


Figure 5. Cumulative Reward. The MAB algorithm used with Baseline, Category, k-Means, and LDA, are UCB2(0.0), Bayes UCB, EG(0.2), and Bayes UCB, respectively.

between the target user and the recommender system. The LDA clustering approach combined with the Bayes UCB MAB algorithm reaches the best performance through the rounds.

5.5. Reproducibility

The dataset we have used in our experiments is publicly available and can be obtained in [Miller et al. 2003]. All base algorithms used here and our implementations are also freely available. Thus, we emphasize that our results are easily reproducible, which enables future developments and possible improvements over our proposed framework.

6. Conclusion and future work

In this work, we propose a new multi-armed bandit algorithm to deal with the user cold-start problem, which works in two steps. The TL-BANDITS framework is flexible and may be instantiated in different ways depending on the scenario of application. We conducted several experiments using a real-world dataset of movie recommendation. The experimental results show that our framework can leverage the state-of-the-art context-free multi-armed bandit algorithms and yield to significant gains over them. As future work, we intend to investigate other combinations of bandit algorithms for each level of the TL-BANDITS framework. We also believe that an appealing line of research is using characteristics of the users to choose the best cluster.

Acknowledgements

The authors gratefully acknowledge the financial support of CEFETMG; CNPq under Proc. 431458/2016-2; and FAPEMIG under Procs. Universal FAPEMIG, process number APQ-03445-16, and FAPEMIG-PRONEX-MASWeb, Models, Algorithms and Systems for the Web, process number APQ-01400-14.

References

- [Agarwal et al. 2009] Agarwal, D., Chen, B.-C., Elango, P., Motgi, N., Park, S.-T., Ramakrishnan, R., Roy, S., and Zachariah, J. (2009). Online models for content optimization. In *Advances in Neural Information Processing Systems*, pages 17–24.
- [Auer et al. 2002] Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.

- [Blei et al. 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *JMLR*, 3(Jan):993–1022.
- [Bouneffouf et al. 2012] Bouneffouf, D., Bouzeghoub, A., and Gançarski, A. (2012). A contextual-bandit algorithm for mobile context-aware recommender system. In *NIPS*, pages 324–331.
- [Efron 1979] Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The annals of Statistics*, pages 1–26.
- [Griffiths and Steyvers 2004] Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *National Academy of Sciences*, 101(suppl 1):5228–5235.
- [Jain 2010] Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.
- [Kaufmann et al. 2012] Kaufmann, E., Cappé, O., and Garivier, A. (2012). On bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 592–600.
- [Leroy et al. 2010] Leroy, V., Cambazoglu, B. B., and Bonchi, F. (2010). Cold start link prediction. In *ACM SIGKDD*, pages 393–402.
- [Li et al. 2010] Li, L., Chu, W., Langford, J., and Schapire, R. (2010). A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670.
- [Mary et al. 2014] Mary, J., Preux, P., and Nicol, O. (2014). Improving offline evaluation of contextual bandit algorithms via bootstrapping techniques. In *ICML*, pages 172–180.
- [Miller et al. 2003] Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., and Riedl, J. (2003). Movielens unplugged: experiences with an occasionally connected recommender system. In *ICIUI*, pages 263–266.
- [Park et al. 2006] Park, S.-T., Pennock, D., Madani, O., Good, N., and DeCoste, D. (2006). Naïve filterbots for robust cold-start recommendations. In *ACM SIGKDD*, pages 699–705.
- [Schein et al. 2002] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *ACM SIGIR*, pages 253–260.
- [Tang et al. 2013] Tang, L., Rosales, R., Singh, A., and Agarwal, D. (2013). Automatic ad format selection via contextual bandits. In *CIKM*, pages 1587–1594.
- [Thompson 1933] Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294.
- [Tokic 2010] Tokic, M. (2010). Adaptive ϵ -greedy exploration in reinforcement learning based on value differences. In *KI 2010: Advances in Artificial Intelligence*, pages 203–210. Springer.
- [Vermorel and Mohri 2005] Vermorel, J. and Mohri, M. (2005). Multi-armed bandit algorithms and empirical evaluation. In *ECML*, pages 437–448.

An Architectural Approach for Developing Intelligent Personal Assistants Supported by NELL

Luís Felipe F. C. Tomazini¹, Oscar J. Romero², Estevam R. Hruschka Jr.^{1,2}

¹Departamento de Computação – Universidade Federal de São Carlos (UFSCar)
13565-905 – São Carlos – SP – Brazil

²Machine Learning Department – Carnegie Mellon University (CMU)
15213 – Pittsburgh – PA – USA

tomazini@ufscar.br, oscarrr@andrew.cmu.edu, estevam@dc.ufscar.br

Abstract. *The growing use of mobile and wearables devices has opened a broad range of possibilities for building robust Intelligent Personal Assistants (IPA's). Most known IPA's implementations have predominantly focused on understanding what the user's immediate intention is, however, there is no evidence of trying to understand the user's context and intentions in the long term in order to provide a real personalization to the user needs. This paper considers the use of a Never-Ending Language Learner (NELL) as an underlying mechanism for IPA's to make decisions and sophisticate rules previously created by the user. We propose an approach to combine NELL's knowledge base to an IPA, handling the knowledge about the world, the user preferences and the device itself.*

1. Introduction

Machine Learning is a research field that grew a lot on the past decades. Nowadays, it is used for many applications, such as spam filtering, bank fraud detection and natural language processing. However, the advances on systems capable of learning on a cumulative way did not follow that pattern. Systems with the ability to use previous information to enhance its learning skills in a never-ending process are rare today, being NELL (Never-Ending Language Learner) the first of this kind to exist [Carlson et al. 2010a]. NELL has been learning to read the web since 2010 and the development of the project continues as can be seen at <http://rtw.ml.cmu.edu>.

NELL's research group has been able to come up with initial results for a system capable of continuously learning in an autonomous, self-supervised and self-reflexive way. Currently, some of NELL's features are: noun phrase classification by categories [Carlson et al. 2010b], noun phrase pairs classification by relations [Carlson et al. 2010b], new relations discovery to extend the initial ontology [Mohamed et al. 2011], inference over the knowledge base to help populating it [Lao et al. 2011, Gardner et al. 2014], polysemy and synonymy resolution [Krishnamurthy and Mitchell 2011], among others. Some of the next steps are to improve NELL's ability to micro read single sentences [Yang and Mitchell 2016, Nakashole and Mitchell 2015], assign temporal scope to NELL's facts [Wijaya et al. 2015], make NELL a conversational agent [Pedro and Hruschka Jr 2012], among others. Partial results on this line of research show the viability of a never-ending learning system [Mitchell et al. 2015].

Considering the context of the never-ending learning system NELL, one of the gaps to be filled is related to the ability of the system to work as a personal assistant

virtual agent, which builds long term relationships with the user. To have this skill the virtual agent must have some cognitive modules, such as, immediate memory, long-term memory about the user and the world, selective attention, feedback systems and many others.

For that reason, the research group decided to combine multiple systems, in order to build an IPA (Intelligent Personal Assistant) with those features. NELL currently has a knowledge base with over 80 million confidence-weighted beliefs (e.g. *animalEatsVegetable(chicken, corn)*); therefore, NELL is capable of providing a large amount of world general knowledge to an IPA. This interaction is beneficial to both systems, as the queries from the IPA will contribute to improve NELL's learning capabilities.

This paper is divided into the following sections: NELL Architecture, An Intelligent Personal Assistant supported by NELL and System Integration. Thus, firstly, NELL's architecture will be explained for a better understanding of the project. Then, the architecture of the intelligent personal assistant will be described. Finally, the integration of both systems will be discussed and analyzed.

2. NELL Architecture

NELL (Never-Ending Language Learner) is a system capable of continuously reading the web indefinitely. Since 2010, NELL uses previous experiences to improve its performance on reading the web, by learning new facts and expanding its knowledge base. This section will describe some of the key components that make it possible for NELL to cumulatively learn new instances.

2.1. Subsystem Components

NELL is built of many subsystem components, which are different semi-supervised algorithms that help it to learn new facts, in a multi-view and multi-task learning approach [Carlson et al. 2010b]. To give a brief overview on NELL's main characteristics, let's focus on a subset of three views and on a subset of two tasks involved in NELL's learning approach.

For the two tasks, consider the classification of noun phrases into categories (Named-Entity Classification, for example *city(Pittsburgh)*) as the first task (**FirstTask**), and the classification of pairs of noun phrases into relations (Relation Classification, example *mayorOf(Bill_Peduto,Pittsburgh)*) as the second task (**SecondTask**). Regarding the three different views, consider, the first one (**FirstView**) to be based on contextual patterns of sentences (such as: *is the mayor of X*, for **FirstTask**, and *X is the mayor of Y*, for **SecondTask**). For the second view (**SecondView**), consider morphological features such as affixes and capitalization. For the third view (**ThirdView**) consider HTML patterns found in web pages read by NELL.

As aforementioned, NELL's different components are based on different algorithms and views, applied to different tasks. The components Coupled Pattern Learner (CPL), Coupled Morphological Classifier (CMC), and Coupled Set Expander for Any Language (CSEAL) all are used in both the **FirstTask**, as well as in the **SecondTask**, but they are based on different views. In this sense, CPL is based on the **FirstView**, CMC is based on the **SecondView**, whereas CSEAL is based on the **ThirdView**.

CPL learns from contextual patterns (e.g. “*inhabitants of X*”, “*Y River in the heart of X*”) to find instances of categories and relations. The Coupled Set Expander for Any Language (CSEAL) queries the Internet with sets of beliefs from each category or relation, it stores lists and tables to extract new instances. In order to filter the lists and tables, it uses mutually exclusive relationships to provide negative examples. The Coupled Morphological Classifier (CMC) has a set of binary classifiers, one per category, classifying candidate facts extracted by other subsystem components according to their morphological features, like parts-of-speech, capitalization and affixes.

2.2. Knowledge Base and Knowledge Integrator

NELL’s knowledge base (KB) is constantly growing and it is used on both reading and learning tasks. On the reading task, the information extracted from the internet is stored in the KB. On the reading task, algorithms try to find inter-related knowledge that can be used as descriptors to help learning new facts.

The initial knowledge base characterizes an ontology, i.e. a collection of predicates defining categories and relations and a set of seed examples for the predicates [Carlson et al. 2010a]. The instances added to the KB are divided into candidate facts and beliefs, depending on their probability. The subsystem components can propose new candidate facts by consulting the internet or even the KB. They do also assign a probability to the candidate facts and the evidences that support it. After that, the Knowledge Integrator (KI) promotes candidate facts to beliefs based on the probabilities assigned (by each one of the system’s components) earlier.

These steps are repeated iteratively. Firstly, each subsystem component runs on the current KB. Then, the KI determines which candidate facts will be promoted. This loop is important for NELL to be able to increase the size of its knowledge base, by increasing its performance on the process of reading the web at each new iteration.

2.3. Macro and Micro Reading

Initially, NELL was designed to perform macro readings [Mitchell et al. 2009], i.e. it is given a considerable amount of texts as input, in this case the internet, and it is expected as output a set of facts mentioned in the given texts, being not necessary that each fact is fully expressed. There is also the micro reading approach, in which a single document (text fragment) is given as input and gets as output complete information about that document. NELL started running (in 2010) based first on Macro-reading mainly because on the first technique it is not required to extract every information from the text and also because in such a large text collection, as the internet, there is a lot of redundancy in information and the macro reading can benefit from this circumstance by combining evidences of multiple sources, leading to a greater reliability of the facts learned. However, such integration that was proposed on the beginnings of NELL, becomes even more necessary nowadays: the cooperation between NELL’s own macro and micro reading, as well as with other systems (such as an Intelligent Personal Assistant, for instance).

It is important to understand the architecture of the micro readers inside NELL, before analyzing their integration with the macro reader. There is a set of specialized micro readers working simultaneously on the process of learning new facts. Each of them has a specific task on this process [Nakashole and Mitchell 2014]. This modularization

of the process enables NELL to have components to make negation detection, named entities extraction, anaphora resolution, and others. This list can be extended indefinitely, depending on the level of comprehension of the language that one expects. There is also a unified micro reader, which uses the annotations made by the specialized micro readers and returns the most likely interpretation to the text given as input.

At this moment, the micro readers work in parallel to better extract the content of texts. There are micro readers to identify events, relations between words and named entities. In order to perform this task, extraction techniques based on the context and also on the morphological structure of the sentence are being used. For example, the syntactic disambiguation micro reader tries to identify, in sequences of the form $\langle verb \rangle \langle noun1 \rangle \langle preposition \rangle \langle noun2 \rangle$, which morphological class is modified by the preposition.

3. An Intelligent Personal Assistant supported by NELL

In this section we describe an IPA which lives on a mobile phone and uses NELL to represent knowledge about the user preferences, the world and the device itself. The IPA continuously builds a NELL-based user model which is organized into three big categories: People that user uses to be in contact with (e.g., family, friends, co-workers, etc.); Activities that user likes to do (e.g., going to the gym, going to work, etc.); and Places that user uses to visit and stays (e.g., home, work place, cinema, etc.). The knowledge stored and maintained by NELL supports the decision-making process performed by the IPA. Figure 1 shows the overall architecture of the IPA.

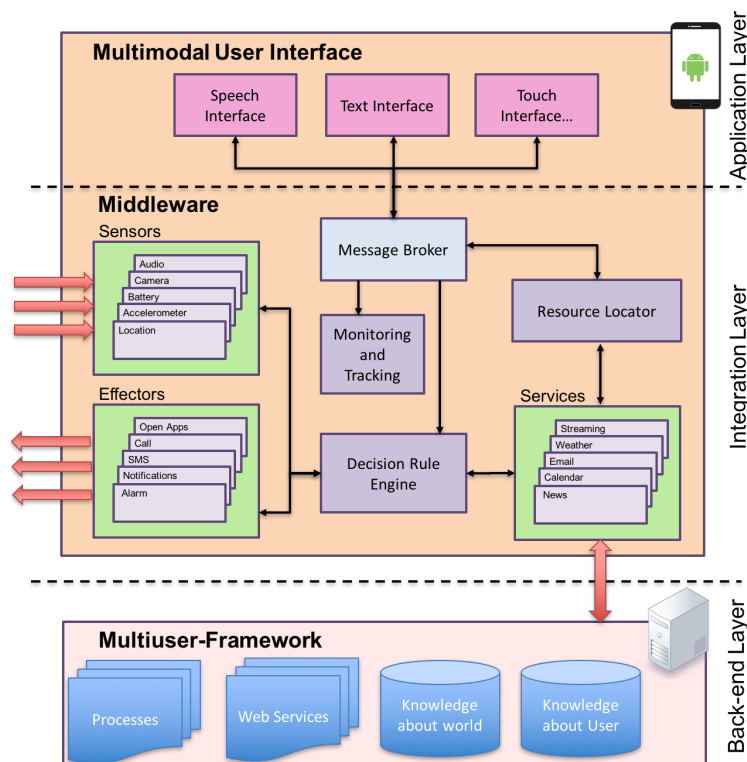


Figure 1. Overall Architecture

The architecture is composed of three main layers: Application, Integration and

Back-end layers. The Application and Integration layers run on the client side, that is, an android-based mobile application, whereas the Back-end layer is deployed on the server side (Amazon Web Services). The Application layer consists of a set of multi-modal interfaces that allows the user to communicate with the IPA through different mechanisms such as speech recognition, text, and touch screen interfaces that process a wide range of gestures. The Integration layer contains a Middleware that allows access to different kinds of sensors, effectors and services. The Middleware also performs some kind of decision-making, as it will be explained in detail later. Finally, the Back-end layer contains all kind of services that must be executed in the server side due to its computational load, for instance, recommendation systems, natural language understanding, NELL processing, among others.

3.1. Middleware

The main purpose of the Middleware is to “glue together” or mediate between components of the Application layer and the Back-end layer. The Middleware layer provides core services like concurrency, transactions, threading, messaging, web services, authentication, among others. This layer also hides (encapsulates) all the complexity involved when communicating with external services. For instance, the communication with a News Reader implies to handle aspects such as: concurrency, synchronous and asynchronous calls, data transformations, execution of background threads, login, cache memory, synchronization, cloning objects, etc.

Generally speaking, the Middleware will contain a Message Broker and a set of sensors, effectors and services. Message broker is the entry point for IPA’s components to communicate with each others, with android services, and with repositories/functionalities hosted in servers. Middleware Services (NELL-Knowledge Service, News Service, Authentication Service, etc.) usually are android-based services that provides an API to communicate with other components and a dedicated cache memory for performance purposes. Therefore, Middleware’s responsibilities can be summarized as follows:

1. to provide access to set of well-defined services (i.e., calendar, weather, news, search, hotel and flight reservations, audio/video streaming, activity recognition, etc.)
2. to provide access to a set of sensors and effectors (location, accelerometer, communication, battery, gyroscope, send sms, send emails, etc.)
3. to mediate communication amongst apps through a Message Broker component which validates, transforms, routes and aggregates all kind of messages that are sent over the apps.
4. to track all the user interaction with apps and system components.
5. to give support to high-level decision-making processes carried out by the IPA.

In the following session we describe how the decision-making process is performed by the IPA using a Decision Rule Engine and NELL.

3.2. Decision Rule Engine

The Decision Rule Engine (DRE) is in charge of validating, storing, processing, executing and specializing rules created by the user. DRE is composed of multiple modules that interact with each other in order to execute an action given a set of specific conditions that

are true. These modules are interconnected as shown in figure 2. The following outlines work flow, message passing and how DRE components cooperate over time to manage rules created by user.

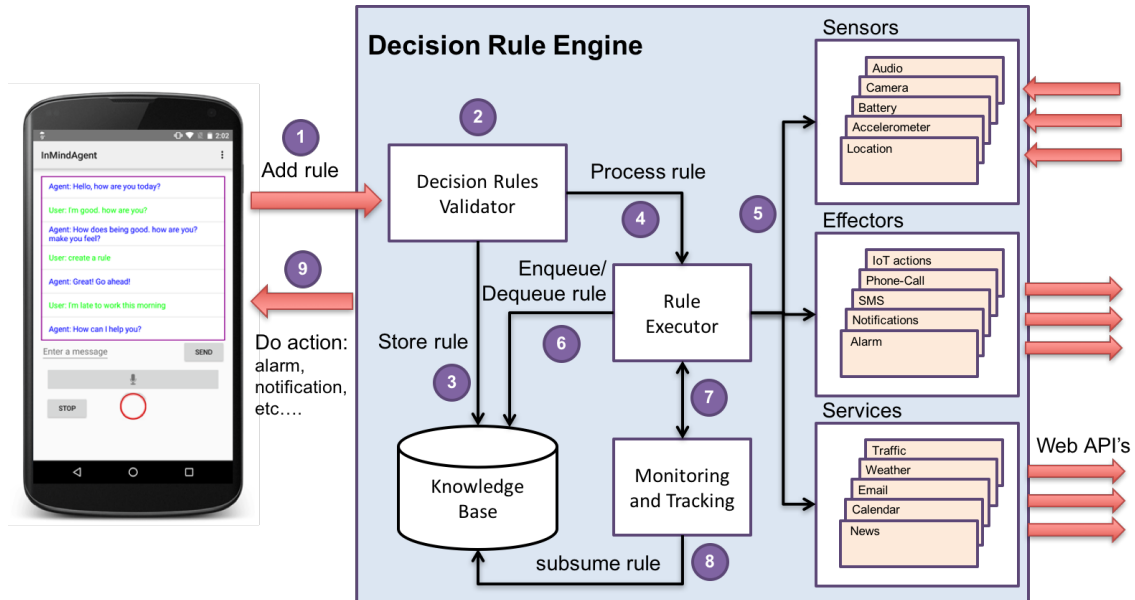


Figure 2. Decision Rule Engine

Scenario - trying to be in time on early morning meetings: Let's assume that the user always is late on meetings that start before 10am. Thus, the user decides to create a rule that triggers a reminder the day before in order to be on time. The rule would look like this:

```
IF:      [CALENDAR.event.Meeting] WHEN [TOMORROW] AND
        [CALENDAR.event.Meeting.StartTime] EARLIER_THAN [DAY.10AM]
THEN:    [ALARM.setReminder] AT [TODAY.8PM]
```

This rule is processed by the DRE as follows:

- **Decision Rule Validator:** after user has defined the rule above (step 1), this component validates the syntax of that rule according to sensor's and effector's attributes and constraints (step 2), for instance, the rule has a condition that refers to attribute [Calendar.event.Meeting.StartTime], so the validator will parse this condition and check that in fact there exists a sensor called *Calendar* that has an attribute called [event.Meeting.startTime], which must be of type *Date* and whose value must be a date/time that occurs later than current date/time.
- **Knowledge Base:** Once the rule is parsed and validated, it is stored on a knowledge base where can be accessed anytime by any component (step 3). These rules are stored locally for performance and privacy reasons, so user's sensitive information is always protected.
- **Rule Executor:** After validation, the rule is immediately processed in order to determine whether it should be executed in that moment (step 4). If so, it will invoke actions from effectors (step 5), otherwise it will add the rule to a queue so it can be executed later when all its conditions are true. Enqueued rules are validated periodically for execution and this

validation may occur at different timing periods (step 6). For example, while a *Weather* rule condition such as <if it is snowing early in the morning> can be checked once every 24 hours, an *Accelerometer* rule condition such as <if the phone is dropping towards the floor> must be validated every 100 milliseconds or so. Other rule conditions such as *Calendar.events* (e.g. <if I have a meeting tomorrow before 10am>) may be validated immediately after the rule is created and then checked again every hour or so (in case new meetings have been added during that time window). Likewise, the actions may have different execution timings, for example, some of these actions can be executed immediately after the conditions are met and some other must be scheduled for posterior execution, for instance, the action <show me a notification right now> is executed right after the conditions are fulfilled, whereas the action <send me a reminder tonight at 10:00 pm> is scheduled for further execution.

- **Monitoring and Tracking:** this module is responsible for monitoring the rule execution process (step 7) by checking whether there exist rules that either are never triggered or are conflicting, for instance, if there are two rules that are triggered at the same time but their actions are opposite (e.g., one rule intends to turn the gps on while the other one intends to turn it off.) When this kind of situation happens, the Monitoring/Tracking module temporarily subsumes the less relevant rule (i.e., the one that has been activated less frequently) and then user is asked to confirm this subsumption decision (step 8).
- **Sensors, Effectors and Services:** sensors and effectors are phone's built-in input and output controls. Sometimes, a control may play both roles as sensor and effector, for instance, the SMS control behaves as sensor when it receives an incoming message and as an effector when it sends out a message. Sensors always match rule conditions whereas Effectors always match rule actions. We are also using *Services* to extend far beyond the set of sensors and effectors available on a smart phone, so they are called cyber-sensors&effectors. These services use REST API's to upload, extract and collect information from web servers such as Weather forecast, News feeds, etc. When and how these Sensors, Effectors and Services interact with each other is determined through the decision rules we have described above. Finally, user is always aware of action execution (i.e., an effector or cyber-effector) through notifications, text messages, alarms, etc. (step 9).

3.3. Automatic and Incremental Improvement of Decision Rules

Allowing the users to create rules that capture the behaviors they would like their phones to exhibit is not a new contribution, existing mobile applications such as IFTTT (IF This Then That, ifttt.com) allows end-users to create simple IF-THEN rules that link a set of triggers (e.g., sensors and web services) with specific actions (e.g., effectors and services). However, IFTTT lacks of a mechanism that makes sure that the rule created by the user really accomplishes the goal for which it was created. In contrast, our approach proposes that the rules created by the user can be automatically improved over time by using NELL along with a generalization/specialization mechanism as explained below. In the following, we outline how the rule sophistication process works. Consider the scenario “trying to be in time on early morning meetings”:

- After rule creation, the Decision Rule Validator (DRV) starts monitoring the new rule to check whether the goal is achieved (be in time on early morning meetings).
- The monitoring system realizes that user continues being late on meetings, so the incremental improvement process is started in order to specialize the rule by identifying patterns of sensors co-occurrence.

- Using NELL as its knowledge base, the DRV infers that the attribute *place* on the calendar event (*Calendar* sensor) is semantically equivalent to attribute *location* used by the *GPS* and *Location* sensors and by the *Traffic* service.
- The DRV realizes that, in order to accomplish the goal of being in time on early meetings, the rule must be specialized, taking the attribute *time* as the observed variable. The DRV finds that attribute *time* is also present in other services, more specifically, NELL can tell the DRV that attribute *trip-duration*, which is an output of *Traffic* service, is measured in *time* units, so the rule should be specialized by adding a new condition that minimizes the *trip-duration* output from *Traffic* service. The *Traffic* service has two location inputs: user's current location and destination location. As a consequence, the DRV infers that *GPS* and *Location* sensors can be used together to estimate the inputs for the *Traffic* service, which in turn will give the system an estimated trip duration for moving from current location to destination.
- If traffic is heavy then *trip-duration* will be increased, so user may be on time if attribute *trip-duration* is also taken into account by the rule. So the DRV specializes the rule by something like:

```

IF :      [CALENDAR.event.Meeting] WHEN [TOMORROW] AND
          [CALENDAR.event.Meeting] EARLIER_THAN [DAY.10AM] AND
          [TRAFFIC.STATUS] IS [TRAFFIC.HEAVY]
THEN:    [ALARM.setReminder] AT [TODAY.8PM] AND
          [TEXT] SET [Traffic is heavy, wake up earlier]

```

- The DRV continues monitoring the rule and realizes that the user is still late on early meetings, so it asks NELL what other calendar attributes can be semantically related to other sensors and services. It discovers that *Weather* sensor has as inputs *location*, *date*, *time*, etc., and it infers that the concept *bad-weather* may worsen the traffic as it increments the trip duration, so again the rule is specialized by something like:

```

IF :      [CALENDAR.event.Meeting] WHEN [TOMORROW] AND
          [CALENDAR.event.Meeting] EARLIER_THAN [DAY.10AM] AND
          [TRAFFIC.STATUS] IS [TRAFFIC.HEAVY] OR
          [WEATHER.STATUS] IS [WEATHER.BAD]
THEN:    [ALARM.setReminder] AT [TODAY.8PM] AND
          [TEXT] SET [You MUST wake up earlier tomorrow]

```

4. System Integration

This section will explain how the information retrieval from the knowledge about the world works. More specifically, it will be described how it connects to NELL. It will be demonstrated how some services benefit from this interaction with NELL.

4.1. WEB API

The services communicate to NELL through a WEB API. There are two distinct ways to make the queries: one is by macro readings and the second is by micro reading. The main difference is that the micro reader returns information based on the context of the words and the macro reader does not take it into account. Because of that, the macro reader needs one entity (a word) to function, while the micro reader needs a complete sentence.

NELL's knowledge base will be consulted, when the queries are made through the API's. Firstly, NELL will try to find the answer on the KB. If the information is there, it will return it. Otherwise, it will try to find the answer outside the KB. It will use its subsystem components to learn more about that topic alongside of other data resources, such as text corpora or the internet. Finally, NELL will store this new knowledge on the KB and return it to the service. Figure 3 shows how the integration through the WEB API's work.

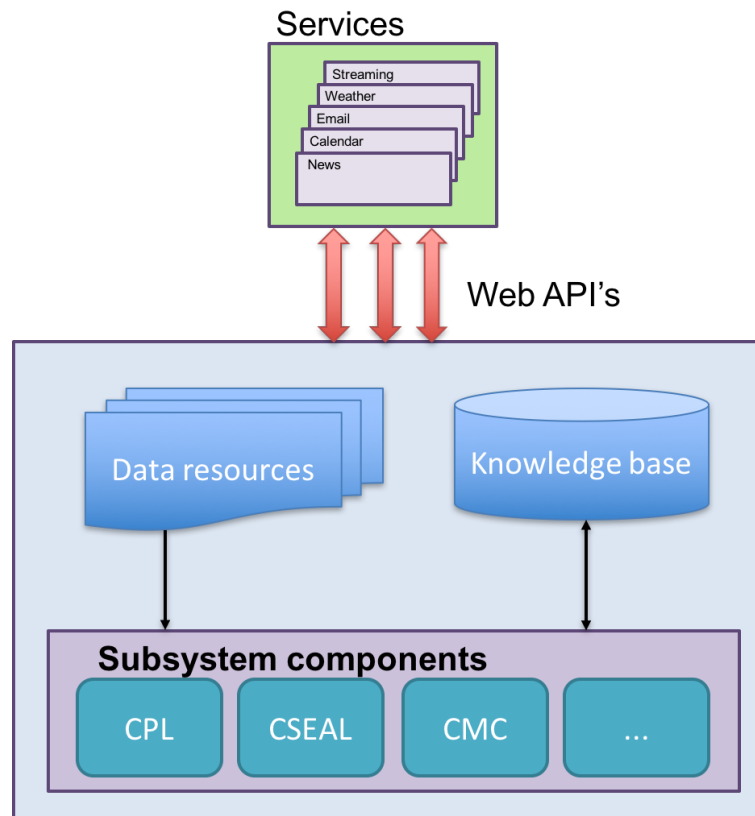


Figure 3. Integration between the services and NELL

4.1.1. Macro Reading

Macro reading allows the middleware to get general facts about the world. These facts can be in the form <Who is Barack Obama?> or <Where is São Paulo?>, for example. In these scenarios, the services will use the WEB API's to query the back-end layer. To perform the macro reading only one instance is necessary, in such examples they would be <Barack Obama> and <São Paulo>. These instances will be queried on NELL to get three different kinds of information: categories, extraction patterns and relations.

The API has three parameters for the queries: *entity1*, *predicate* and *entity2*. In order to get the most likely categories that *entity1* belongs the query uses only two arguments, *entity1* is the instance, *predicate* is * as the intention is to get all the possible categories for that instance and *entity2* is left blank in this case.

Using the same parameters as on the previous query, it is possible to get the reasons used to learn those categories. These reasons come from the Coupled Pattern Learner (CPL), that is one

of NELL's subsystem components. These reasons do not provide direct answers for the IPA, but they give information about the context that the instance occurs.

If three parameters are used, the query will return the most likely relationships. In this case *entity1* is the instance, *predicate* is * and *entity2* is * as the intention is to get all the possible relations for that instance. The API will check for all the relations that the instance belongs, but will display only the ones with a confidence of 75% or more. In a similar approach as used in the categories, with the same parameters, the API returns the extraction patterns for the relations.

4.1.2. Micro Reading

In micro reading, a text fragment is given as input and the output is complete information about the text. The information comes in the form of annotations that consist of a confidence score (not necessarily a probability), the annotator and the value. The value is the entity found in the text and the categories or relations it belongs.

The API has only one parameter: the sentence to micro read. In one query, it is possible to retrieve both the categories and relations found in the text. Firstly, the categories are extracted and they are returned alongside of their confidence score and the entity. Then, it looks for categories for each permutation of the instances found before.

The micro reader is useful for services such as the news service. It is possible to micro read the news headlines to get some annotations. They might be in the context of noun phrases, verbs (for relations), proper nouns, events, and even for sentiment analysis. After that the IPA can recommend other news articles that would be interesting for the user, by creating rules based on the annotations.

5. Discussion

Both systems benefit from the integration: the IPA receives the answers of the knowledge about the world and the user, while NELL is able to expand its ontology, by learning new facts. In this section, it will be explained in detail how each system gain from the union of them.

NELL is important to the intelligent personal assistant as it is used to represent knowledge about the user preferences, the world and the device itself. This knowledge supports the decision-making process of the IPA. In the step of monitoring the rule execution process NELL is crucial to identify whether there are conflicting rules created by the user and to subsume the least relevant one.

In some categories, it is possible that NELL will not learn new facts, as it is close to the point that there is nothing else to learn. This never-ending learner does not have self-reflexive components to guide it and focus on other topics. For NELL to have such a feature it must know which categories and relations are under-performing. This is a second key point for the integration of both systems, because queries made by the IPA can subsidize an improvement on NELL's capabilities to learn.

This collaboration with other systems became essential to NELL. Nowadays, NELL reads without specific focus. There are many predefined categories and it learns in order to populate them. However, sometimes it does not learn new facts about a specific topic. Such circumstances might arise for two reasons: either the learning algorithm is not the most suitable for that field or the system is close to its learning limit on that topic, such as with country names. With the integration of both systems NELL will have the ability to be self-reflexive and can have a focus when it is learning, being able to target at different categories, which still have a lot of information

to be collected. To exemplify, if some queries about a specific topic were made by the IPA and NELL could answer only a few or none of them, then NELL's learning capabilities on that topic are not satisfactory, and it should focus more on that category, to expand the knowledge base. Going back to the countries example, if queries about countries were made and all or many of them were answered, then it means that NELL has a good understanding of that subject and can concentrate on other categories.

With the proposed integration, NELL can be closer to the way humans learn. If we are reading a book about colors and we do not understand the meaning of many words that are important to the comprehension, we consult a dictionary to learn more. Such a natural process for us is what we want to bring to NELL. The first part refers to NELL's reading abilities, as the second to the queries made by the IPA and the dictionary is analogous to the internet. In this way, NELL will have an auto reflexive and guided learning through the queries made by the IPA and NELL's learning algorithms.

Acknowledgements

This work was supported in part by the Program Science without Borders (Programa Ciência sem Fronteiras) from CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil and by Yahoo! through the CMU – Yahoo InMind project.

References

- Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr., E. R., and Mitchell, T. M. (2010a). Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*.
- Carlson, A., Betteridge, J., Wang, R. C., Hruschka Jr., E. R., and Mitchell, T. M. (2010b). Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10*, pages 101–110, New York, NY, USA. ACM.
- Gardner, M., Talukdar, P., Krishnamurthy, J., and Mitchell, T. (2014). Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar. Association for Computational Linguistics.
- Krishnamurthy, J. and Mitchell, T. M. (2011). Which noun phrases denote which concepts. In *Proceedings of the Forty Ninth Annual Meeting of the Association for Computational Linguistics*.
- Lao, N., Mitchell, T., and Cohen, W. W. (2011). Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 529–539, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Mitchell, T., Betteridge, J., Carlson, A., Hruschka Jr., E. R., and Wang, R. (2009). Populating the semantic web by macro-reading internet text. In *The Semantic Web (ISWC 2009)*, pages 998–1002.
- Mitchell, T., Cohen, W., Hruschka Jr., E. R., Talukdar, P., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N.,

- Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., and Welling, J. (2015). Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.
- Mohamed, T., Hruschka, E., and Mitchell, T. (2011). Discovering relations between noun categories. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1447–1455, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Nakashole, N. and Mitchell, T. M. (2014). Micro reading with priors: Towards second generation machine readers.
- Nakashole, N. and Mitchell, T. M. (2015). A Knowledge-Intensive Model for Prepositional Phrase Attachment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 365–375.
- Pedro, S. D. and Hruschka Jr, E. R. (2012). Conversing learning: Active learning and active social interaction for human supervision in never-ending learning systems. In *Advances in Artificial Intelligence—IBERAMIA 2012*, pages 231–240. Springer.
- Wijaya, D. T., Nakashole, N., and Mitchell, T. M. (2015). "A Spousal Relation Begins with a Deletion of engage and Ends with an Addition of divorce": Learning State Changing Verbs from Wikipedia Revision History. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Yang, B. and Mitchell, T. (2016). Joint extraction of events and entities within a document context. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Towards Crime Rate Prediction through Street-level Images and Siamese Convolutional Neural Networks

Virginia Ortiz Andersson, Marco Antônio Ferreira Birck,
Ricardo Matsumura Araújo and Cristian Cechinel

¹PPGC – Federal University of Pelotas (UFPel)
Rua Gomes Carneiro, 1 - Centro - CEP 96010-610 – Pelotas – RS – Brazil

{vandersson, mafbirck, ricardo}@inf.ufpel.edu.br, cris16marte@yahoo.com.br

Abstract. *The analysis of the environment for crime prediction is based on the premise that criminal behavior is influenced by the nature of the environment in which occurs. Street-level images are the closest digital depiction available of the urban environment, in which most street crimes take place. This work proposes a crime rate prediction model that uses street-level images to classify street crimes into four crime rates levels (from low to high). For that, we use a 4-Cardinal Siamese Convolution Neural Network (4CSCNN) and train and test our analytic model in a region of Chicago, US, that showed high street crime concentrations between the years of 2014 and 2015. With this experiment, we investigate the use of convolutional neural networks (CNN) for the task of crime rating through visual scene analysis and found possibilities towards automatic crime rate predictions using CNN models.*

1. Introduction

Several works have been developed to understand the factors that trigger a criminal event carried out by an individual, as well as the risks involved and the measures to avoid it. Over the years, theories have been developed to map criminals behavior and crime itself. “Environmental theories” consider crime as a confluence of *offenders, targets* and specific *laws and settings* at *particular times and places* [Wortley and Mazerolle 2008] [Brantingham and Brantingham 1995]. In here, offenders are not the central object of interest, but one element of a *crime event*.

According to [Wortley and Mazerolle 2008], the environmental perspective is based on the premise that criminal behavior is influenced by the nature of the environment in which occurs, i.e., the environment plays a fundamental role in initiating crime and shaping its course. The distribution of crime in time and space is non-random and it depends on environmental factors and situations. Control and crime prevention are then a result of understanding the role of the environment on crimes patterns. Environmental theories can focus on (i) how offenders react in the environment, such as the *Routine Activity* [Cohen and Felson 1979], the *Rational Choice* [Brantingham and Brantingham 1993], and the *Crime Pattern Theory* [Eck and Weisburd 1995], and (ii) how to map the physical environment in which the criminals operate, such as the well known *Broken Windows Theory* [Wilson and L. 1982] and the *Routine Active of Places* [Sherman et al. 1989].

Street-level images are the closest depiction of the human environment available in digital form, and their use in daily life is gradually increasing, mostly to aid navigation. The Google Street View [Google 2017] service popularized the access to such

street images and computer vision models are being used together with street-level images to relate a city's physical appearance with crime statistics [Arietta and Efros 2014] [Khosla et al. 2014] [Gebru et al. 2017].

In this work, we present an initial study on crime rate prediction models that use street-level images as input. We propose a 4-Cardinal Siamese Convolution Neural Network (4-CSCNN) together with a Multi-layer Perceptron to classify visual scenes depicted in street-level images into four categories of crime rate, from low to high rates. We built a dataset with street-level images from a region of Chicago City, US, that showed high street crimes rates between the years of 2014 and 2015. This region was divided into equal sized cells containing the total crime events that happened inside the cell region. Images belonging to the cell received the label according to the total crimes in that region. We use this street-level images dataset to train and test our proposed model, obtaining 54.3% of overall accuracy and 77% of average accuracy per class, in the classification of the four crime rate categories. The achieved results are discussed in Section 4.

This paper is organized as follows. Section 2 presents background and related work to Crime Prediction Models, Visual Scene and Environment Analysis and Siamese Convolutional Neural Networks. Section 3 describes how data was collected together with the methodology followed by our approach. Section 4 discusses the most important findings of our work, and Section 5 presents some conclusions and ideas for future work.

2. Related Work

Our paper is related to (i) Crime Prediction Models, that are the main purpose of this work, (ii) Visual Scene and Environment Analysis, which inspired the use of visual attributes in our proposed model and (iii) Siamese Convolutional Neural Networks, the deep learning technique which is used in our model.

2.1. Crime Prediction Models

Based on *Routine Activity of Places Theory* [Sherman et al. 1989], crime *hot spots* are regions where high concentration of crime events is observed. The *hot spots* were initially used as a criminal data visualization technique, and further became a prediction model, with the advance of statistical and geographic information (GIS) tools, and the support of the observed characteristics of crime events *repeat* and *near-repeat* [Sherman et al. 1989]. *Hot spots* were used by [Block 1998], in the *Early Warning System*, that used data collected by the community and law enforcement to produce the *hot spots* and point out possible near regions affected by violent crimes.

In [Bowers et al. 2004], authors proposed the use of a technique to distribute crime in a geographic surface and calculate risk assessment of crime events - the prospective risk surface - to obtain *hot spots*. This technique consists of using a two-dimensional grid with n equally sized cells overlying the geographic region of interest. In the model, a weight is associated with geographically located crimes. Recent crimes that happened near the center of a cell receives a higher weight. The weights of all crimes near the center are added together to produce the risk index of the belonging cell. To evaluate the proposed model, the authors used “theft” data from Merseyside County, England, from the year of 1997. The model presented 62% to 64% of accuracy in predicting crimes for a 2-day time window.

Later, [Chainey et al. 2008] proposed the use of *Kernel Density Estimation* (KDE) to map *hot spots* overlaying a geographic area. It smooths the criminal data over a region according to a kernel density estimator function, mapping probabilities for a crime event happening under a specific area. In [Johansson et al. 2015], the authors evaluate the KDE approach reaching 76% to 84% of accuracy in predicting crimes in a 3-month time window. More recently, [Gerber 2014] proposed a modification in KDE estimator to hold independent variables. The author used *tweets* from a specific location in Chicago city, Illinois, correlated with crime events in the location.

Risk Terrain Modeling (RTM) was proposed by [Caplan et al. 2011] to assess crime risk over a region. RTM consist of acquiring crime-related factors and standardize each factor to a common geographic region, usually assigning a weight to the presence or absence of this factor at every place covered by the region of interest. As the risk value over a region gets higher, the probability of a crime event occurring in that region also gets higher. According to [Caplan et al. 2011], the RTM technique produces maps that indicate regions with greater risks of becoming *hot spots* in the future. In [Drawve et al. 2016], the authors propose an aggregate neighborhood risk of crime (AN-ROC) measure, applying RTM model to forecast neighborhood-level of violent crimes in Little Rock, Arkansas. They identify 14 risk factors, such as banks, bus stops, check-cashing, convenience stores, fast-food restaurants, grocery stores, hotel/motels, liquor stores, pawn shops, tattoo/piercing shops that were expected to influence crime. The AN-ROC measure was obtained by averaging the risk of crime per cell by neighborhood.

2.2. Visual Scene Environment Analysis

Computer vision and machine learning are used extensively to discover environment attributes in street-level images. According to [Dubey et al. 2016], visual scene and urban imagery analysis can be focused on different objectives, such as predicting perceptual responses from images, understanding cities by their visual urban scene, understanding the connection between urban appearance and socioeconomic factors and rank or compare different urban environments. The strains which the present work is interested in are understanding cities and the connection between urban appearance and socioeconomic factors.

In [Doersch et al. 2012], the authors proposed a methodology to automatically find visual elements (e.g. windows, balconies, street signs) from street-level images that are distinctive for a specific geographic area, i.e., they occur much more frequently in that area than other areas. The authors used Google Street View, a street-level image database, and clustered images using Histograms of Oriented Gradient (HOG) and color components to compose descriptors for specific patches (e.g. windows, signs, doors). They clustered the patches using Nearest Neighbor algorithm, dividing positive and negative data clusters into l equally-sized subsets. Next, they iteratively trained an SVM detector for each visual element using the detectors trained in the previous iteration in a new unseen subset, selecting top k detection for retraining. As results, for Paris, the authors achieved 83% of accuracy (where chance yields 50%) and for Prague City, 92% of accuracy. Later, [Arietta and Efros 2014] followed the proposed model of [Doersch et al. 2012] and applied Clustering and Support Vector Regression (SVR) to USA cities, discovering predictive relationships between visual elements from the environment and non-visual variables like crime and theft rates, housing prices, population density, graffiti density and percep-

tion of danger. They compared the use of HOG+color descriptors with the fifth convolutional layer of Caffe’s ImageNet CNN model and concluded that HOG+color descriptors were more visually consistent but captured less city semantics.

In [Khosla et al. 2014], the authors explored the ability to use visual scenes to predict distances of surroundings establishments such as hospitals and fast-food restaurants. They also explored the possibility of predicting crime rates in an area using visual scenes. The authors experimented different descriptors, e.g. GIST, Texture using Local Binary Pattern (LBP), Color using Locally-Constrained Linear Coding (LLC), HOG+Color and FC7 layers from Caffe’s ImageNet CNN model. They used an SVR machine on the image features obtained by each descriptor. The results achieved in finding hospitals and fast-food restaurants showed similar results between Color, HOG+color and Deep Learning descriptors, varying from 0.58% to 0.61% of accuracy. To predict crime rates and danger perception, the authors used HOG+color descriptors and the SVR machine and compared to a human test prediction. The results achieved for crime rate prediction was an accuracy of 72.0%, compared to 59.6% in human tests.

2.3. Siamese Convolutional Neural Networks

In [Bromley et al. 1993], Siamese Neural Networks are defined as “twin” Neural Networks (NNs) which share their parameters. The main reason to use twin NNs is that the inputs will be mapped to a very near space since they are being processed by the same function e.g. Convolutional Neural Networks (CNNs). The Siamese NNs are widely used to discriminate and match tasks - e.g. [Bromley et al. 1993],[Taigman et al. 2014], [Lin et al. 2015], [Zagoruyko and Komodakis 2015]. Recently, focusing in city places safeness classification through street-level images, [Dubey et al. 2016] proposed the Place Pulse 2.0, a novel dataset containing a pairwise comparison of “safe” and “unsafe” images from 56 cities across the world. The authors also proposed two different Siamese CNN architectures: Streetscore-CNN (SS-CNN), to predict a winner between two images comparison as safe and unsafe, and a Ranking Streetscore-CNN (RSS-CNN), where a ranking function is attached to the SS-CNN to rank the street-level images into high to low safeness scores.

A 4-Cardinal Directions pseudo-Siamese Convolutional Neural Network was proposed by [Lieman-Sifry 2016], where the author used CNNs to extract a low-level representation of images from geographical points in Colorado, USA state, aiming at determining the location of the images. The CNNs architecture used was based on ResNet [He et al. 2015] and AlexNet [Krizhevsky et al. 2012], reserving two networks per cardinal direction, using the method of ensembles in conjunction. The network learns “from scratch” to classify regions from Colorado state. The representations mapped by the author’s CNNs for each cardinal direction are concatenated and passed through two fully connected Multi-layer Perceptron layers (MLP) that classify the representation according to the given labels.

3. Methodology

We present in this work a model to classify crime rates in a city region, using only visual attributes extracted from street-level images from specific locations. The main objective of the model is, given a georeferenced location (point), predict the historical crime density

in that location based on the images that surround it, in the four cardinal orientations. We intend to understand how the visual scene influences the crime rate in each place.

3.1. Crime Data

The crime data acquired for this work was retrieved from Chicago City data portal [CityOfChicago 2017]. The dataset contains records of crime events georeferenced since 2001. We chose to focus on crime data between in the year of 2014 and 2015. To visualize and identify crime hot spots in Chicago, we create a grid-like data structure of shape 50 x 50, dividing the limits of the city into 2.500 equally spaced cells and distributed the crime records that happened in the street or sidewalk into the corresponding cells, using latitude and longitude. Figure 1(a) shows crime events that happened in “street” or “sidewalk” in the year 2014 grouped by a cell. Darker shades of blue are used to denote that at least one crime occurred in that area, and more crimes per cell are indicated by lighter and hotter shades of blue, green, orange and red. We choose the area with higher crime rate concentration, highlighted with a red square in Figure 1(a), for our study.

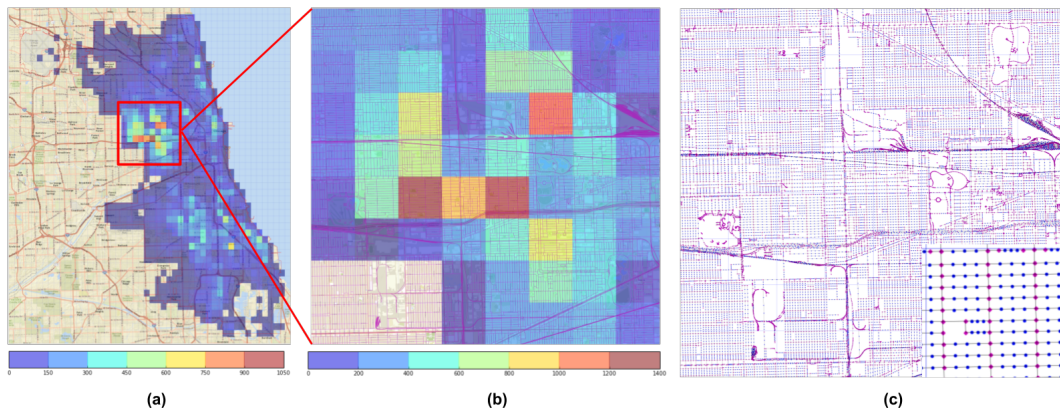


Figure 1. (a) Hot spots of street and sidewalk crimes in Chicago, in the year 2014. Crimes are distributed in a 50 x 50 grid, with low to high concentrations depicted in the color scale, showing total crime range by color. The region with more concentrated crimes between neighbors cells is highlighted with a square. In (b), streets of the region of interest with street and sidewalk crime hot spots in the year of 2014. The points that shape the streets of the selected area in Chicago are presented in (c), with the street corners depicted (lower right detail) as diamonds and the interpolated points as circles.

3.2. Obtaining and Labeling Street-level imagery

From the selected region in Figure 1(a), we create a new grid data structure of shape 8 x 8 with 64 equally spaced cells, corresponding to the region of interest in the original grid. Next, we obtain a *shapefile* from streets distribution of the selected region using Mapzen Metro Extracts service [Mapzen 2017], superimposing the *shapefile* vectors on the map, as depicted in Figure 1(b). The *shapefile* of Chicago streets contains 2D points, or vertices, that shape the lines of streets. We transformed the vertices into latitude and longitude coordinates and interpolated points between the vertices, as depicted in Figure 1(c). With Google Street View API [Google 2017] we obtained 4 images, from the vertices and interpolated points that shape the streets, corresponding to the 4 cardinal directions. Each point with their 4 images was distributed by grid cell. Next, each cell

received a label according to the crime distribution in the region of interest. We categorized the distribution into “blue”, “green”, “orange” and “red” to indicate low to high street crime rates.

Using the methodology described, we built a street-level imagery dataset with a total of 20,056 points, with 80,226 images for each point. Each point belongs to a specific cell and a specific label according to the crime rate in the cell region. To build the train and test dataset, a number of georeferenced distinct points were randomly chosen for each label. Table 1 shows the total number of points for each label - blue, green, orange and red - and total number of images in the train set. The test set was adjusted to have the same number of points based on the smallest number of examples a label could have.

Table 1. Image dataset composition arranged by labels, in train and test set.

Label	Train set		Test set	
	Points	Images	Points	Images
Blue	3,128	12,512	1,073	4,292
Green	4,128	16,512	1,073	4,292
Orange	4,327	17,308	1,073	4,292
Red	4,181	16,724	1,073	4,292
Total	15,764	63,058	4,292	17,168

3.3. Proposed Siamese CNN

Our work uses as inspiration the conjunction of ideas from [Zagoruyko and Komodakis 2015] and [Lieman-Sifry 2016]. While [Lieman-Sifry 2016] uses a variant of a Siamese network, as [Zagoruyko and Komodakis 2015] stated, this method can be named as a Pseudo-Siamese, because the CNNs used are free to learn specific features to each image subset. Our proposed model, however, uses the concept of Siamese Neural Network in which all the CNN have their weights “frozen”, and the previous weights domain reused. Figure 2 shows the proposed Siamese CNN architecture, showing the 4-cardinal CNNs with shared and frozen weights trained with *ImageNet* and fully connected Multi-layer Perceptron (MLP) layers (FCs). The CNNs follows the AlexNet architecture [Krizhevsky et al. 2012], making our model more robust compared to the simpler CNN architecture used in [Lieman-Sifry 2016]. Each CNN is frozen, following a common approach of *transfer learning*, allowing the use of pre-trained weights from the *ImageNet* domain. This way, the model is less likely to overfit and has the ability to leverage knowledge from the *ImageNet* domain.

In the proposed model architecture, the resulting CNNs outputs, a smaller spatial representation of the images, are inputted in the *Fully Connected FC 1024* MLP units that are free to learn. Next, the 4 outputs are concatenated and passed through another classifier, an MLP that has been proved to be very efficient in conjunction with CNNs. The MLP has two fully connected layers, the *FC1 4096* and *FC2 4*, where the backpropagation algorithm [Rumelhart et al. 1988, LeCun et al. 1998] is applied, and the output is passed to a *softmax* classifier to extract the distributed probability over the labels.

3.4. Experiments

The proposed model was trained using Keras [Chollet 2015] and Tensorflow [Martín 2015] on a NVIDIA graphic processing unit (GPU) with 8GB of memory. As

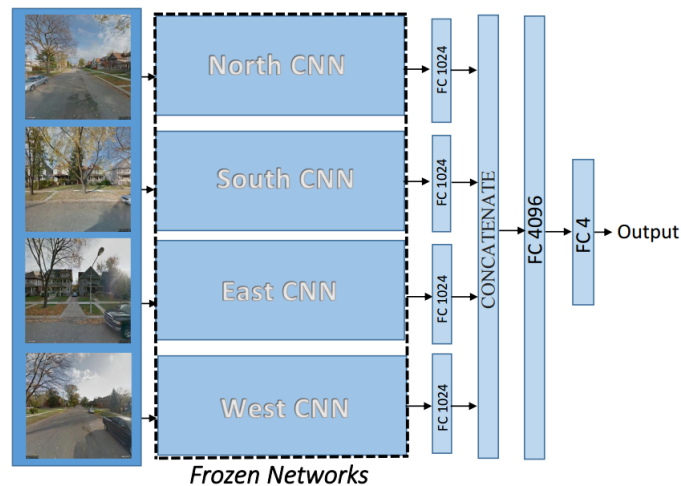


Figure 2. Proposed architecture with 4-cardinal Siamese CNNs, with Street View images following 4 cardinal orientations, using frozen *ImageNet* weights and fully connected layers followed by a *softmax* classifier.

the CNNs layers were frozen, the training process was faster compared to training the CNNs “from scratch”, allowing only the fully connected (FC) layers to be trained. The loss function chosen was categorical *cross-entropy*, due to the multi-label nature of our problem. The *adaDelta* optimizer [Zeiler 2012] was used to train the model, with the ability of automatically adapt the learning rate. To improve generalization, we select 32 random images of our image dataset and applied a data augmentation method, consisting in random (i) image cuts with $range = 0.2$ i.e. the shear angle in counter-clockwise direction as radians, (ii) image zooms, with $zoom - range = 0.2$ in a random mode defined as $[lower, upper] = [1 - zoom - range, 1 + zoom - range]$ and also (iii) horizontal flips, helping our model to generalize better and deal with the data bias.

In the training step, our model receives as inputs $n^i \in N$, $e^i \in E$, $s^i \in S$ and $w^i \in W$, where N, E, S, W are subsets of images from our dataset composed of Google Street View images, taken each from the cardinal direction North, East, South and West respectively. Additionally, i is a geographical point of a street that belongs to a cell with a labeled crime rate. Each one of these four images was passed through the model in batches of 32 geographic points for the mini-batch training. Each image linked to one cardinal direction was passed through the respective CNN to obtain a representation in a low dimensional space, a descriptor with deep features of the image. All image descriptors are then concatenated to represent a new low-dimensional descriptor of the whole geographical point. Furthermore, the resulting descriptor is passed through the fully connected MLP layer whose objective is to map the geographic point to one of the four crime rate labels.

Training was executed with a maximum of 600 epochs, and checkpoints in 150, 360 and 600 epochs were set to observe possible overfitting. In the MLP we applied an aggressive *dropout* [Hinton et al. 2012] to avoid overfitting due to the high entropic capacity of the model. We turned off 75% random units in the *FC 1024* and *FC 4096* layers, to prevent overfitting, forcing the model to learn with less architectural capacity, about 25% of the total units.

With the purpose of analyzing the results from the proposed 4-CSCNN, we built a simplified model, replacing the CNNs by Histogram of Oriented Gradients (HOG) descriptors [Dalal and Triggs 2005] and the fully-connected layers by a simple MLP classifier. We resize each image to 128x128 pixels and applied the HOG methodology using patches of 8x8 pixels, with 8 directions and 1 cell per block. Each cardinal HOG resulted in a feature dimension of 2048. Similar to the proposed 4-CSCNN model, the features were concatenated, with the resulting dimension of 8192. Next, the MLP algorithm was trained to classify each location point into one of the 4 crime rate labels. The baseline MLP was built with 2 hidden layers, with 4 and 2 neurons respectively, using the *stochastic gradient descent* optimization algorithm and 200 epochs. We train and test the baseline model 10 times, shuffling the datasets each time, and averaged the overall accuracy to obtain a score metric for further comparison.

4. Results and Discussion

As depicted in Figures 3(a) and 3(b), high values during training opposing lower values in validation or test step indicates overfitting, which was observed after 150 epochs in our model. The results obtained during training step are presented in Figure 3(a), including the loss value for each epoch. Figure 3(b) shows validation loss values and overall accuracy calculated during validation step. Loss value is the sum of the errors obtained during the evaluation of each example, and it's expected to diminish during training and to be low during validation or test step. In our model, the loss value obtained during validation was approximately 1.

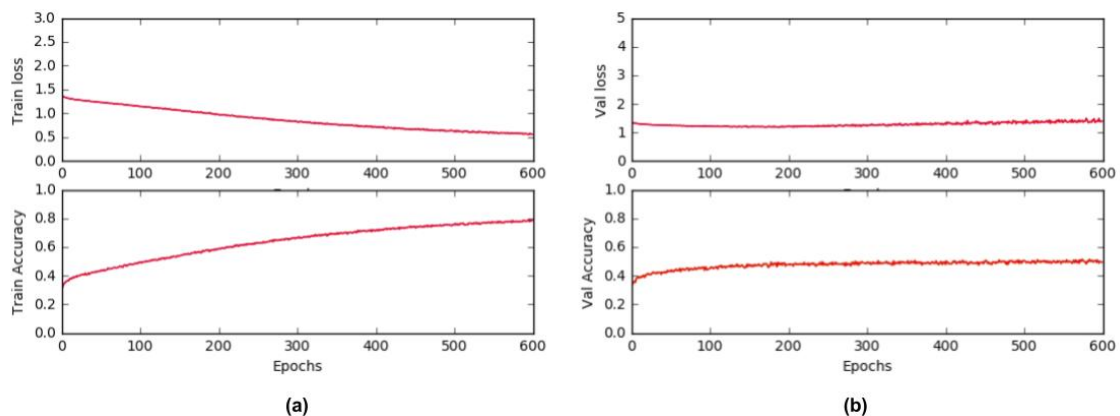


Figure 3. Loss and accuracy values during training (a) and validation (or test) (b) steps.

Table 2 shows the confusion matrix of the results obtained during validation step, at the 150 epoch, right before the model shows signs of overfitting. For each crime rate labels, 1073 location points containing 4 images were validated. From all points tested, about 34.2% of all predictions were classified as “red”, or as a local with high crime rates, 28.3% as “orange”, 21.6% as “green” and 15.7% of all predictions were classified as “blue”, with low crime rates. The presence of fewer examples for low crime rates and the chosen distribution of the labels may explain the tendency of the proposed model to classify most images as medium or high crime rates. The accuracy per label, calculated as one-against-all using the equation $(TP + TN)/(TP + TN + FP + FN)$ is presented

in Table 3. Following the values for each one of the labels, the accuracy per label varied between 73% to 81%, and the average accuracy obtained was approximately 77%. The overall accuracy, calculated as $\sum(\text{mainDiagonal})/\text{population}$, at the 150th epoch was 54.3%.

Table 2. Confusion matrix showing the classification between four labels indicating low to high crime rates resulted from validation step.

Predicted \ Actual	Blue	Green	Orange	Red	Total
Blue	470	144	210	249	1073
Green	97	512	195	269	1073
Orange	57	141	636	239	1073
Red	50	134	176	713	1073
Total Predicted	674	931	1217	1470	4292

Following Table 3, from the total predicted examples as “blue”, 69% of the time the model was correct, i.e., the precision of the model in predicting a local as “blue”, and, from all the times that the model should have predicted a local as having low crime rates, it correctly predicted about 43% of the examples i.e., the recall. The values for high crime rates, or “red”, were about 48% of precision and 66% of recall, due to more examples classified as “red”.

Table 3. Results obtained during testing: accuracy, precision and recall for all classified labels.

	Blue	Green	Orange	Red
Population	4292	4292	4292	4292
TP: True Positive	470	512	636	713
TN: True Negative	3015	2800	2638	2462
FP: False Positive	204	419	581	757
FN: False Negative	603	561	437	360
Accuracy	0.811	0.771	0.762	0.739
Precision	0.697	0.549	0.522	0.485
Recall	0.438	0.477	0.592	0.664
Avg. Label Accuracy	0.77	(77%)		
Overall Accuracy	0.543	(54.3%)		

Table 4 shows the achieved results with the baseline model, compared with the results from the proposed 4-CSCNN network. CNNs need more examples than classic computer vision descriptors, that only needs few examples to achieve reasonable results. The results obtained with the simplified model demonstrate the non-trivial nature of the problem addressed, necessitating the investigation of a model that better serves the classification of the criminal rate through street-level images. Our proposed 4-CSCNN performed better considering the overall accuracy of the simple baseline-MLP model.

Table 4. Comparison between the proposed 4-CSCNN and the baseline model with HOG+MLP classifier.

Model	Overall Accuracy
Baseline-MLP	0.455
4-CSCNN	0.543

5. Conclusion

This paper presents a preliminary study on predicting crime rates from street-level images, which represent the urban environment where street crime occurs. For this, we proposed a new 4-Cardinal Siamese Convolutional Neural Network (4-CSCNN) architecture to predict urban crime rates, given a georeferenced location point. The model uses 4 images surrounding the given point, facing north, south, east and west. Each image is the input of one CNN, with pre-trained frozen weights from *AlexNet* architecture [Krizhevsky et al. 2012]. At the output of each CNN, a *Fully Connected* (FC) layer was attached, and the resulting descriptors were merged into one only descriptor, that is finally classified by a Multi-layer Perceptron (MLP) into one of the four crime rate labels.

The CNNs are responsible for learning features of the environment images which may affect crime rates. The use of 4 images surrounding a location gives more information about the environment than using a single image as input. The obtained results, 54.3% of overall accuracy, indicates that the architecture can infer a possible relation between environment features and crime rates, using only street-level images. The CNNs achieved better results when compared with the baseline model with HOG descriptors, considering the overall accuracy score. Although the result obtained for overall accuracy is less than the related works, our 4-CSCNN model can be distinguished from related works by differences in how we tackle the input image. Our model does not focus on specific attributes in the image, instead, it uses the whole scene as input, and it is still a preliminary study to investigate the use of CNNs to discover the most important features in images for the crime prediction task.

For future works, we intend to implement techniques of deep visualization in the neurons, to show what image inputs cause higher activation in units. By doing this, we search for the features that the CNNs learned to be related to low or high crime rates. This can be useful for social and law enforcement analysis of the urban environment. Also, our model has potential to be applied to different problems i.e. 4-Cardinal images of one georeferenced point can be related to different statistics and environment characteristics, e.g. city region classification as downtown and suburbs, that requires surrounding visualization.

Acknowledgements

We acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

References

- Arietta, S. M. and Efros, A. A. (2014). City Forensics : Using Visual Elements to Predict Non-Visual City Attributes. *Transactions on Visualization and Computer Graphics*, 20(12):2624–2633.

- Block, C. (1998). The GeoArchive: An information foundation for community policing. *Crime mapping and crime prevention*, pages 27–81.
- Bowers, K. J., Johnson, S. D., and Pease, K. (2004). Prospective hot-spotting: The future of crime mapping? *British Journal of Criminology*, 44(5):641–658.
- Brantingham, P. J. and Brantingham, P. L. (1993). Environment, routine and situation: Toward a pattern theory of crime. *Advances in criminological theory*, 5:259–294.
- Brantingham, P. P. and Brantingham, P. P. (1995). Criminality of place. *European Journal on Criminal Policy and Research*, 3(3):5–26.
- Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E., and Shah, R. (1993). Signature verification using a “siamese” time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688.
- Caplan, J. M., Kennedy, L. W., and Miller, J. (2011). Risk terrain modeling: Broker-ing criminological theory and gis methods for crime forecasting. *Justice Quarterly*, 28(2):360–381.
- Chainey, S., Tompson, L., and Uhlig, S. (2008). The Utility of Hotspot Mapping for Predicting Spatial Patterns of Crime. *Security Journal*, 21:4–28.
- Chollet, F. (2015). Keras. <https://github.com/fchollet/keras>.
- CityOfChicago (2017). Chicago data portal. <https://data.cityofchicago.org/>.
- Cohen, L. E. and Felson, M. (1979). Social Change and Crime Rate Trends: A Routine Activity Approach. *American Sociological Review*, 44(4):588–608.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, volume I, pages 886–893.
- Doersch, C., Singh, S., Gupta, A., Sivic, J., and Efros, A. a. (2012). What makes Paris look like Paris? *ACM Transactions on Graphics*, 31(4):1–9.
- Drawve, G., Thomas, S. A., and Walker, J. T. (2016). Bringing the physical environment back into neighborhood research: The utility of RTM for developing an aggregate neighborhood risk of crime measure. *Journal of Criminal Justice*, 44:21–29.
- Dubey, A., Naik, N., Parikh, D., Raskar, R., and Hidalgo, C. A. (2016). Deep learning the city: Quantifying urban perception at a global scale. pages 196–212.
- Eck, J. E. and Weisburd, D. L. (1995). Crime Places in Crime Theory. *Crime Prevention Studies*, 4:1–33.
- Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Aiden, E. L., and Fei-Fei, L. (2017). Using Deep Learning and Google Street View to Estimate the Demographic Makeup of the US. pages 1–41.
- Gerber, M. S. (2014). Predicting crime using Twitter and kernel density estimation. *Decision Support Systems*, 61(1):115–125.
- Google (2017). Google Street View API. <https://developers.google.com/maps/documentation/streetview>. [Online; accessed 09-May-2017].

- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.
- Johansson, E., Gahlin, C., and Borg, A. (2015). Crime Hotspots: An Evaluation of the KDE Spatial Mapping Technique. In *EISIC European Intelligence and Security Informatics Conference*, pages 69–74, Manchester, UK. IEEE.
- Khosla, A., An, B., and Lim, J. (2014). Looking beyond the visible scene. *2014 IEEE Conference on*.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Lieman-Sifry, J. (2016). Convolutional neural networks to predict location from colorado google street view images: Galvanize capstone project. <https://github.com/jliemansifry/streetview/>.
- Lin, T.-Y., Cui, Y., Belongie, S., and Hays, J. (2015). Learning deep representations for ground-to-aerial geolocation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5007–5015. IEEE.
- Mapzen (2017). Mapzen metro extracts. <https://mapzen.com/data/metro-extracts/>. [Online; accessed 09-May-2017].
- Martín, Abadi, e. a. M. A. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.
- Sherman, L. W., Gartin, P. R., and Burger, M. E. (1989). Hot spots of predatory crime : routine activities and the criminology of place. *Criminology*, 27(June):27—55.
- Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.
- Wilson, J. Q. and L., K. G. (1982). Broken Windows. *Atlantic Monthly*, 249(3):29.
- Wortley, R. and Mazerolle, L. (2008). *Environmental criminology and crime analysis: situating the theory, analytic approach and application*.
- Zagoruyko, S. and Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361.
- Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Uma abordagem Fuzzy-Adaptativa para Segmentação de Imagens de Cromossomos

Maria F. S. Andrade¹, Filipe R. Cordeiro¹, Valmir M. Filho¹,
Fabiana F. Lima², Suy F. Hwang² e Julyanne C. G. Mendonça²

¹Departamento de Estatística e Informática (DEINFO)
Universidade Federal Rural de Pernambuco (UFRPE)

²Divisão de Técnicas Analíticas e Nucleares (DITAN)
Centro Regional de Ciências Nucleares do Nordeste (CRCN-NE/CNEN)

{fsandrade25, filipe.rolim, valmir.ufrpe}@gmail.com,

{fflima, shwang}@cnen.gov.br, july_cgm@yahoo.com.br

Abstract. *Chromosome analysis is an important task to detect genetic diseases. However, the process of identifying chromosomes can be very time consuming. Therefore, the use of an automatic process to detect chromosomes is an important step to aid the diagnosis. The proposed work develop a new approach to automatize the segmentation of chromosomes, using an adaptive thresholding combined with fuzzy logic. The proposed method is evaluated using the database from CRCN-NE, which has 35 images. Results showed that the proposed approach compared with state of the art techniques obtained better segmentation results, with sensitivity and specificity values of 91% and 92%, respectively.*

Resumo. *A análise de cromossomos é uma tarefa importante para detecção de doenças genéticas. No entanto, o processo de identificar cromossomos demanda muito tempo. Portanto, a utilização de um processo automático de detecção de cromossomos é importante para auxiliar na análise. O trabalho proposto propõe uma abordagem híbrida de segmentação de cromossomos, utilizando limiarização adaptativa combinada com lógica fuzzy. O método proposto é avaliado para a base de dados do CRCN-NE, que consiste de 35 imagens de cromossomos. Resultados mostram que a abordagem proposta comparada com trabalhos do estado da arte conseguiu obter melhores resultados de segmentação, com taxas de sensibilidade e especificidade de 91% e 92%, respectivamente.*

1. Introdução

Os cromossomos estão presentes em todas as células nucleadas de todos os organismos vivos, tendo como função levar informações genéticas para a reprodução das células e organismos [Pantaleão et al. 2003]. A análise e classificação de cromossomos pode identificar várias anomalias associadas à alterações na estrutura dos cromossomos, tais como a síndrome de Down, Turner [Cunha et al. 2015] e na identificação de vários tipos de câncer [NUSSBAUM et al. 2008]. No entanto, esse processo é muito custoso tanto para o citogeneticista, quanto para o paciente, que demora para receber o tratamento adequado. Para realizar o diagnóstico, o citogeneticista precisa analisar centenas de imagens para selecionar aquelas com os cromossomos em estado de metáfase para posterior análise [Grisan et al. 2009]. Posteriormente, o citogeneticista analisa cada imagem

com os cromossomos, realizando a contagem e identificação de cada um. Essa análise é geralmente realizada de forma manual em muitos hospitais, sendo esse um procedimento trabalhoso e que demanda muito tempo [Arora and Dhir 2016]. Além disso, esse processo normalmente resulta em divergência de diagnósticos para a mesma imagem [Khmelninskii et al. 2008]. Isso acontece porque algumas imagens, obtidas através de microscópios, geralmente são de baixa qualidade, tornando o processo manual impreciso [Kurtz 2011].

Além do tempo necessário para a análise de um único indivíduo, os cromossomos apresentam diversos tipos de estruturas, podendo estar sobrepostos, encurvados ou apresentar algum tipo de anormalidade. Muitos trabalhos têm sido desenvolvidos para realizar a análise automática de cromossomos, mas apesar das pesquisas realizadas, o processo de segmentação e classificação automática de cromossomos ainda é um problema em aberto [Arora and Dhir 2016].

O processo de segmentação automática de cromossomos tem dois impactos principais: primeiramente, o processo irá reduzir o tempo de seleção manual do cromossomo na imagem, e por consequência, irá economizar tempo de processamento e análise de imagens de cromossomo para propósitos de contagem e classificação. O trabalho proposto visa desenvolver um algoritmo para realizar a segmentação automática de cromossomos, fornecendo uma ferramenta de auxílio ao médico, permitindo um diagnóstico mais rápido e preciso.

2. Trabalhos Relacionados

Alguns trabalhos têm sido desenvolvidos para auxiliar na automatização de análises cromossômicas. Esses trabalhos se dividem em três tipos: técnicas de pré-processamento, segmentação e classificação [Arora and Dhir 2016].

Kurtz *et al.* [Kurtz 2011] aplica técnicas de processamento de imagens, como suavização, realce de contraste, dilatação e limiarização para segmentar os cromossomos visando auxiliar o citogeneticista na análise da forma do cromossomo, não abrangendo o tratamento de imagens com cromossomos tortos ou sobrepostos. No trabalho de Matta [Matta 2013], foi desenvolvida uma ferramenta computacional, denominada ChromoSomeClassification, que auxilia na quantificação da dose de radiação absorvida por um indivíduo, através da contagem de aberrações cromossômicas, utilizando uma rede neural *Radial-Basis Functions* (RBF) [Haykin 2001] para identificá-los. Porém, ele também não trata imagens de cromossomos sobrepostos, sendo em alguns casos necessária a separação manual.

Minaee *et al.* [Minaee et al. 2014] utilizaram critérios geométricos para identificar os cromossomos que se tocam e se sobrepõem parcialmente, encontrando uma linha de corte e separando-os. Madian *et al.* [Madian et al. 2015] propõem um algoritmo para identificar a zona de sobreposição de cromossomos na imagem sem nenhuma intervenção humana, utilizando funções de curvatura para identificar os pontos de corte e realizar a separação, conseguindo fazê-la com uma acurácia de 96% de acerto. Poletti *et al.* [Poletti et al. 2012] também criaram uma metodologia de segmentação de cromossomos que se tocam e se sobrepõem gerando, para cada conjunto de cromossomos sobrepostos, uma árvore com possíveis formas para separá-los.

Uttamanin *et al.* [Uttamanin et al. 2013b] propuseram o método MetaSel, que

classifica o cromossomo baseado em 8 características geométricas. O MetaSel realiza a segmentação e classificação de cromossomos sobrepostos. No entanto, a técnica possui um alto custo computacional. Em outro trabalho, Uttamanin [Uttamanin et al. 2013a] propôs um método de classificação que divide as imagens em alta e baixa resolução. A baixa resolução é usada para contagem dos cromossomos, enquanto que as imagens de alta resolução são usadas para encontrar anomalias estruturais.

3. Método Proposto

Um dos métodos mais utilizados para segmentação de cromossomos é a binarização [Poletti et al. 2012, Arora and Dhir 2016]. O método desenvolvido realiza a segmentação utilizando duas técnicas de binarização: adaptativa e *fuzzy*. O fluxograma da abordagem proposta pode ser visto na Figura 1.

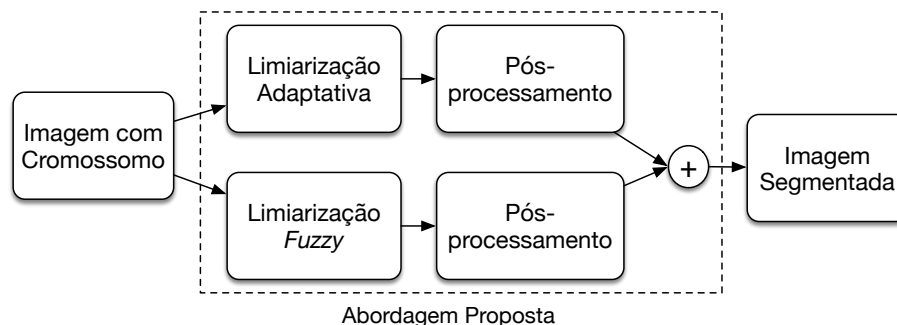


Figura 1. Fluxograma do Método Proposto

Apesar da técnica adaptativa ter sido usada em outros trabalhos [Poletti et al. 2012], a abordagem utilizando lógica *Fuzzy* nunca foi aplicada para esse problema. No modelo proposto, as duas técnicas podem ser aplicadas em paralelo e cada uma delas é responsável por extrair características distintas da imagem. O processo começa com a utilização da imagem de cromossomo, a qual é convertida para tons de cinza para iniciar o processo de segmentação. Cada uma das etapas de segmentação é descrita a seguir.

3.1. Limiarização Adaptativa

O limiar adaptativo é um método de binarização que define um valor de limiar diferente para cada *pixel* da imagem, baseado em valores de sua vizinhança [Burger and Burge 2016]. Desta forma, o limiar se adapta à região que está sendo analisada, conseguindo obter os melhores valores de limiar de acordo com nível de similaridade dos *pixels* ao redor.

Uma das abordagens do método adaptativo é analisar estatisticamente as regiões da imagem e obter um valor de limiar T para cada região. Um dos métodos mais utilizados é calcular a média aritmética ou a média Gaussiana dos *pixels* de cada região. Na média aritmética, cada *pixel* contribui igualmente para o cálculo do limiar. Na média Gaussiana, *pixels* distantes do centro da região contribuem menos para o cálculo do limiar. O cálculo do limiar é feito através da Equação 1

$$T = \text{mean}(I_L) - C, \quad (1)$$

onde *mean* representa a média aritmética ou Gaussiana, I_L é uma sub-região da imagem I e C é uma constante de ajuste do limiar. Para a segmentação utilizando limiarização adaptativa, é definido um tamanho de janela W , que percorre toda a imagem, através do processo de convolução.

3.2. Limiarização Fuzzy

Diferente dos conjuntos clássicos, elementos de um conjunto *fuzzy* tem graus de pertinência em relação àquele conjunto. O grau de pertinência de um conjunto *fuzzy* indica a certeza de que o elemento pertence ao conjunto [Dubois et al. 2014]. Um conjunto *fuzzy* é representado por uma função de pertinência $\mu_A : X \rightarrow [0, 1]$ [Dubois and Prade 1978]. Portanto, para todo $x \in X$, μ_A indica o grau de pertinência de x ao conjunto *fuzzy* A .

Um número *fuzzy* é definido como um tipo de conjunto *fuzzy*, geralmente definido a partir do conjunto de números reais e possui algumas propriedades como: ser normalizado, convexo e outras. A função de pertinência de um número fuzzy gaussiano, que é utilizada nesse trabalho, é definida pela Equação 2:

$$\mu_A(x) = e^{-\frac{(x-c)^2}{\beta}}, \quad (2)$$

onde c e β são o centro e largura do número *fuzzy*, respectivamente. O número x no centro da região gaussiana tem o valor de pertinência máximo e os valores de pertinência da vizinhança decrescem gradualmente.

Quando se aplica lógica *fuzzy* em imagens, os valores de intensidade de cada *pixel* podem ser descritos como um número *fuzzy*, em que é calculado um valor de pertinência relacionado à sua vizinhança. Essa abordagem difere da abordagem clássica porque ele não classifica os *pixels* como fundo e objeto diretamente, mas ele atribui um nível de pertinência à região vizinha. Esse nível de pertinência pode variar entre 0 e 1 e posteriormente um limiar T é aplicado para a função de pertinência para realizar a binarização. Desta forma, um *pixel* com valor de pertinência alta, indica que ele tem valor de intensidade similar à região de vizinhança.

Seja $A_{N \times M}$ uma imagem com $N \times M$ *pixels*, a fuzificação da imagem é feita através da convolução, usando uma janela espacial de tamanho $W \times W$, onde a função de pertinência gaussiana é definida para o centro da região. A função de pertinência define um valor de pertinência para cada *pixel* i, j , definido como $\mu_{p(i,j)}$, usando a Equação 3:

$$\mu_{p(i,j)} = \frac{\sum_{k=1}^W \sum_{l=1}^W e^{-\frac{(A(i,j)-c)^2}{\beta}}}{W^2 - 1}. \quad (3)$$

A largura da função de pertinência pode ser ajustada variando o valor da variável β . Um baixo valor de β implica em menos tolerância à diferença de contraste na vizinhança do *pixel*. Em algumas abordagens, o valor de β pode ser mantido fixo e ajustado apenas o valor de limiar sobre a matriz de pertinência.

3.3. Pós-processamento

Ao final de cada etapa de segmentação, existem ruídos caracterizados por pequenas regiões da imagem, que são removidos através da etapa de pós-processamento. Com

a finalidade de remover essas imperfeições, foi testado dois tipos de operadores morfológicos nas imagens binarizadas: erosão e abertura. Cada abordagem é seguida de uma etapa de remoção de regiões menores que um valor pré-determinado. A análise das duas abordagens de pós-processamento e dos valores dos tamanhos de remoção são avaliados na seção de resultados.

3.4. Composição Fuzzy-Adaptativa

A última etapa da método proposto é a composição das imagens binarizadas pelo método adaptativo e pelo método *fuzzy*. Essa composição é feita através da adição das imagens binarizadas em cada etapa. Em seguida, é verificado quais elementos da imagem composta fazem interseção com elementos da binarização adaptativa. Esse processo é realizado para reduzir ruídos da imagem. O pseudocódigo da composição das imagens pode ser visto a seguir.

Algoritmo 1: FUZZY-ADAPTATIVO

Entrada: I_{adap} , I_{Fuzzy}
Saída: Imagem Segmentada

```

1 início
2    $I_{final} = I_{Fuzzy} + I_{adap}$ 
3    $cntsFinal \leftarrow$  contornos de  $I_{final}$ 
4   para cada  $c \in cntsFinal$  faça
5     caso  $c \cap I_{adap} = \emptyset$  faça
6       | remove  $c$  de  $I_{final}$ 
7     fim
8   fim
9 fim
10 retorna  $I_{final}$ 

```

A etapa de composição se inicia com as imagens I_{adap} e I_{Fuzzy} , que correspondem às binarizações da segmentação adaptativa e *fuzzy*, respectivamente. Em seguida, é feita a operação de adição entre as duas imagens, obtendo-se a imagem final, representado por I_{final} . A partir da imagem final são removidas alguns resíduos da imagem, que foram adicionados no processo de composição. Para isso, são extraídos os contornos de I_{final} e, para cada contorno, é feita uma interseção com a região equivalente em I_{adap} . Caso a interseção seja vazia, significa que uma região não pertencente aos cromossomos foi incluída no processo de composição *fuzzy*-adaptativo. Desta forma, a região é excluída da imagem final. A Figura 2 mostra as diferentes etapas do método proposto.

A Figura 2 (a) mostra uma região onde estão localizados os cromossomos na imagem original. Apesar do método ser aplicado para a imagem inteira, na Figura 2 é mostrado apenas a região dos cromossomos para facilitar a visualização. A Figura 2 (b) apresenta os resultados da etapa adaptativa enquanto que a Figura 2 (c) mostra os resultados da etapa *fuzzy*. A etapa adaptativa consegue segmentar a maioria dos cromossomos. Porém, alguns componentes de cromossomos são perdidos nessa etapa. A etapa *fuzzy* realiza a detecção das bordas dos cromossomos, o que auxilia na identificação da localização dos mesmos. Um preenchimento de região não pode ser utilizado nessa etapa, pois muitos componentes não possuem região fechada. Na etapa final, representada na Figura 2 (d), os elementos encontrados nas etapas adaptativa e *fuzzy* se complementam, recuperando

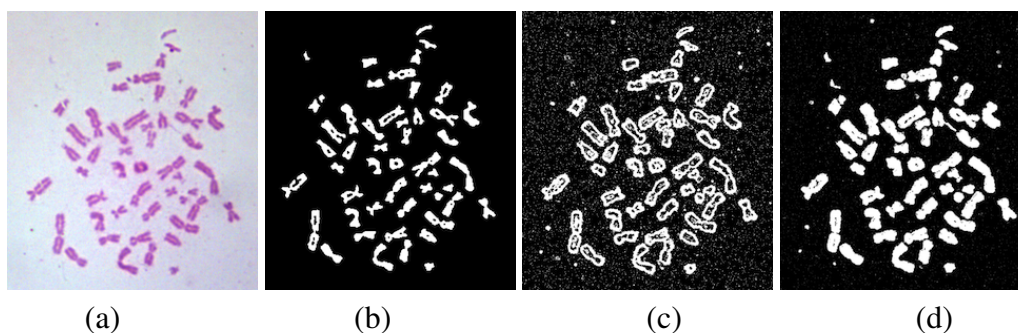


Figura 2. Etapas de segmentação do Método Proposto: (a) Imagem Original; (b) Etapa Adaptativa; (c) Etapa Fuzzy; (d) Composição Final Fuzzy-Adaptativa.

os componentes de cromossomos perdidos. Essa abordagem se difere dos trabalhos da literatura, que aplicam apenas um método de segmentação. Ao utilizar um método híbrido, a composição de técnicas são complementares na segmentação final.

4. Metodologia

4.1. Ambiente Experimental

As imagens utilizadas durante o experimento foram adquiridas no Centro Regional de Ciências Nucleares do Nordeste (CRCN-NE). O CRCN disponibilizou uma das bases de imagens de cromossomos para o desenvolvimento do algoritmo, todas rotuladas por especialista. Atualmente, no CRCN, o processo de análise cromossômica é feito manualmente, ou seja, um especialista por meio de um microscópio óptico analisa ao menos 1000 lâminas por cada amostra irradiada. A base de dados adquirida consta de 35 imagens de cromossomos metafásicos e suas respectivas rotulações. As imagens digitalizadas possuem resolução 2028×1536 pixels.

Os algoritmos implementados nesse trabalho foram desenvolvidos utilizando a linguagem Python juntamente com a biblioteca OpenCV. Os algoritmos foram executados utilizando um computador com processador i7 e 8GB de memória RAM.

4.2. Métricas

Para realizar essa análise entre as técnicas, foram implementadas métricas de análise de imagens médicas, do estado da arte: Sensibilidade, Especificidade, Índice Jaccard e Acurácia Balanceada. Essas métricas são calculadas a partir das taxas de verdadeiro positivo (VP), verdadeiro negativo (VN), falso positivo (FP) e falso negativo (FN) encontradas nas imagens segmentadas. A seguir são descritas cada uma das métricas analisadas.

Sensibilidade (SE) representa a eficácia do algoritmo em classificar corretamente os pixels do objeto da imagem. A Sensibilidade pode ser descrita através da Equação 4.

$$SE = \frac{VP}{VP + FN} \quad (4)$$

A Especificidade (SP) representa a eficácia do algoritmo em classificar corretamente os pixels de fundo da imagem. A Especificidade pode ser descrita através da

Equação 5.

$$SP = \frac{VN}{FP + VN} \quad (5)$$

A Acurácia Balanceada (BAC) é a média da especificidade e sensibilidade. Quanto maior o valor de BAC mais equilibrada é a relação entre SE e SP, tendo um menor desvio padrão entre os valores de SE e SP. A acurácia balanceada é calculada de acordo com a Equação 6.

$$BAC = \frac{SE + SP}{2} \quad (6)$$

O índice de Jaccard (J) representa a taxa de similaridade entre a imagem gerada e a sua respectiva imagem ouro. O Índice de Jaccard é calculado através da equação 7.

$$J = \frac{VP}{VP + FN + FP} \quad (7)$$

O coeficiente de correlação de Matthews (MCC) é considerada uma métrica útil para avaliar a similaridade entre classificações binárias [Poletti et al. 2012]. Essa métrica representa se a predição dos valores está sendo aleatória ou não. Quanto mais perto de 0, mais aleatório é o resultado. Um valor de -1 indica total divergência entre as imagem, enquanto um valor de +1 representa igualdade total. A métrica MCC é calculada através da Equação 8.

$$MCC = \frac{VP \times VN - FP \times FN}{\sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}} \quad (8)$$

O valor de predição positiva (PPV) representa o número total de *pixels* classifica-dos corretamente como objeto, sobre o número total de *pixels* classificados como objeto, conforme mostrado na Equação 9.

$$PPV = \frac{VP}{VP + FP} \quad (9)$$

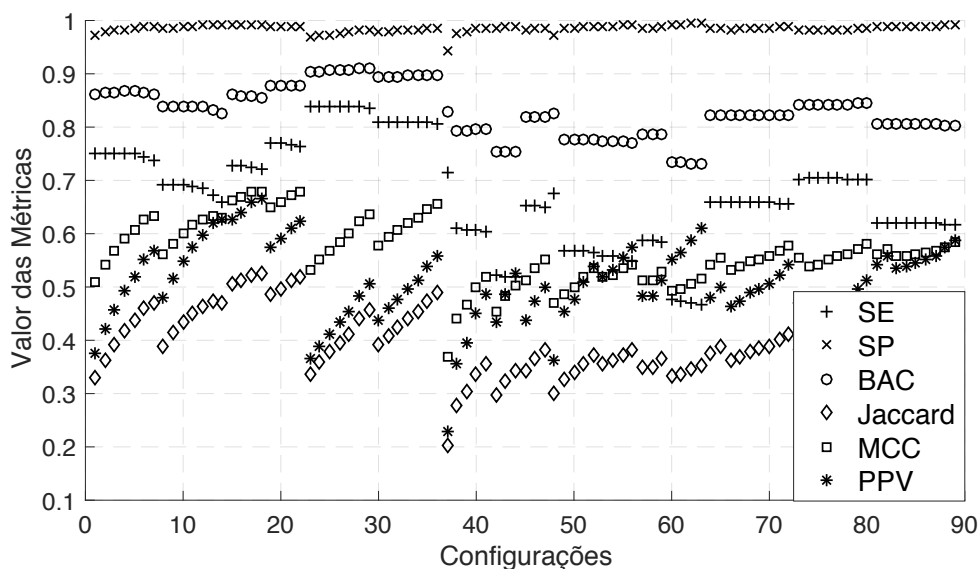
5. Resultados

Inicialmente foi feita uma análise dos parâmetros do método proposto. Para definir os melhores parâmetros, foram utilizadas as métricas descritas na seção 4.2. Para a etapa adaptativa, foram analisados os parâmetros de método adaptativo, tamanho de janela e constante de ajuste, representados por M_{adap} , W_{adap} e C_{adap} , respectivamente. Para a etapa *fuzzy* foram analisados os parâmetros de tamanho de janela e limiar, representados por W_{fuzzy} e T_{fuzzy} , respectivamente. No método *fuzzy*, o valor de β , descrito na equação 3 é mantido fixo, com valor 1, enquanto é variado apenas o limiar para a função de pertinência. Isso acontece porque alterar β ou o limiar tem o mesmo impacto na segmentação. Por fim, também foram explorados os parâmetros de pós processamento, tais como operação morfológica e tamanho máximo dos elementos removidos, representados por PO_{som} e PO_{rem_size} , respectivamente. Os parâmetros explorados na análise realizada são mostrados na Tabela 1.

Tabela 1. Parâmetros explorados para o método proposto

Parâmetros	Configurações
M_{adap}	média, gaussiano
W_{adap}	3 a 21
C_{adap}	3, 4, 5, 6, 7
W_{fuzzy}	5, 7, 9, 1
T_{fuzzy}	0,1 a 0,5
P_{osom}	abertura, erosão
$P_{\text{osrem_size}}$	10 a 50

Foi feito um estudo inicial do impacto da variação de parâmetros na fase de limiarização adaptativa em relação às métricas descritas. A Figura 3 mostra o resultado dessas métricas para 90 variações de configuração da etapa de limiarização adaptativa juntamente com o pós-processamento. Os parâmetros variados nessa análise são M_{adap} , W_{adap} , C_{adap} , P_{osom} e $P_{\text{osrem_size}}$, descritos na Tabela 1.

**Figura 3. Análise de variação de parâmetros para o limiar adaptativo**

Na Figura 3, o eixo x representa as diferentes configurações e o eixo y representa os valores das métricas. Essa análise mostra que as configurações não tem grande impacto sobre a métrica de especificidade, representado por SP. Além disso, é possível observar que as métricas de Jaccard, MCC e PPV, possuem correlação, pois quando um aumenta as outras aumentam. No entanto, a sensibilidade nem sempre melhora quando MCC, PPV ou Jaccard apresentam valores maiores. Pela Figura 3, também é possível concluir que o método adaptativo é sensível à variação de parâmetros para as métricas Jaccard, MCC e PPV, enquanto que para SE e BAC a variação é menor. Com relação aos valores dos parâmetros, foi possível obter os melhores valores usando M_{adap} com método de média, C_{adap} igual a 7, P_{osom} com operador de abertura e $P_{\text{osrem_size}}$ com valor entre 30 e 50.

A Figura 4 mostra a etapa de pós-processamento realizada após a segmentação adaptativa. A Figura 4 (a) mostra a imagem segmentada pela etapa do método adaptativo

e a Figura 4 (b) mostra a imagem resultante após a operação de erosão e remoção de pequenas regiões na imagem. Como pode ser visto na imagem, mesmo após a aplicação do pós processamento, ainda restam na imagem elementos não pertencentes aos conjuntos de cromossomos, os quais poderão ser removidos em uma etapa posterior de classificação.

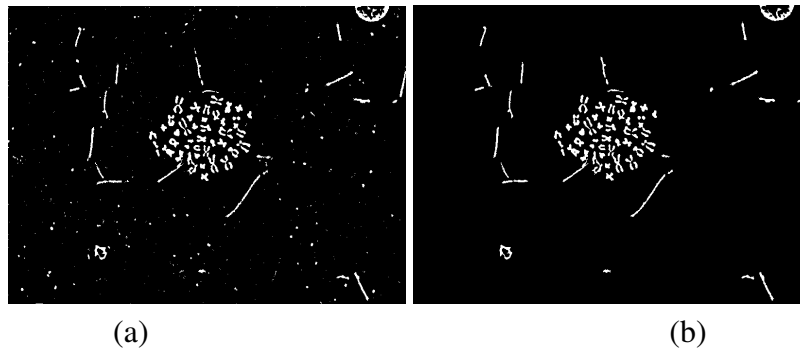


Figura 4. Etapa de pós-processamento no método proposto: (a) imagem segmentada pela etapa adaptativa; (b) imagem após operação de erosão.

A análise de parâmetros também foi realizada para a etapa de segmentação utilizando lógica *fuzzy*. No caso da abordagem *fuzzy*, aplicar os operadores de pós processamento não gerou melhoria nos resultados, já que apenas as regiões de borda dos cromossomos foram detectadas e essas áreas, em alguns casos, apresentam tamanho próximo ao de ruídos externos, o que são removidos na etapa de pós processamento. Desta forma, aplicar o operador de erosão pode remover também o contorno dos cromossomos. A Figura 5 mostra os resultados obtidos para variação de parâmetros na etapa de limiarização *fuzzy*.

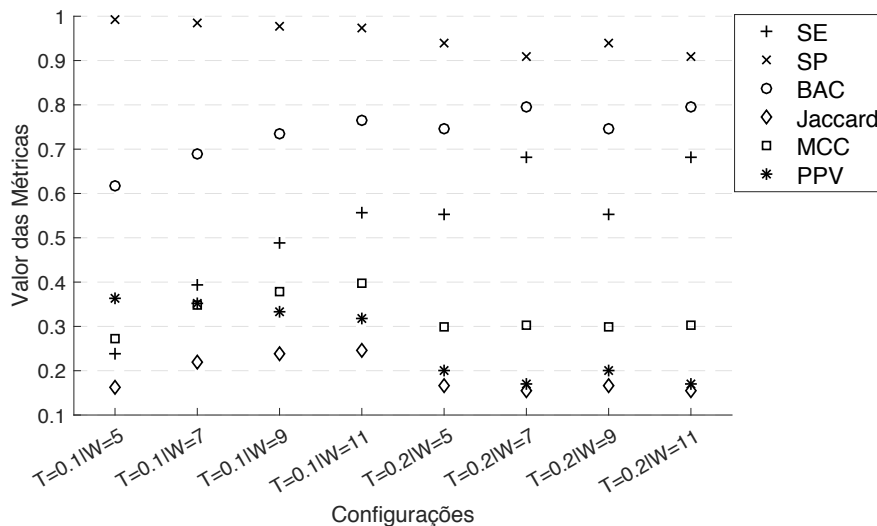


Figura 5. Análise de variação de parâmetros para a binarização utilizando lógica *fuzzy*

Apesar da variação de parâmetros do algoritmo *fuzzy* ter sido realizada para valores de T entre 0,1 e 0,5, a binarização se mostrou sensível para valores de T acima de 0,2, obtendo resultados pouco precisos. Desta forma, na Figura 5 são apresentadas apenas as melhores configurações, com valores de T entre 0,1 e 0,2. Entre os melhores valores

obtidos, está a configuração com $T = 0.1$ e $W = 11$. Conforme descrito anteriormente, a etapa de pós processamento não adicionou melhorias nessa etapa.

5.1. Análise de Segmentação

O método proposto foi comparado com técnicas de segmentação utilizadas no estado da arte. Um dos métodos mais utilizados para segmentação de cromossomos é o método de Otsu [Jianzhuang et al. 1991], sendo usado pelos melhores trabalhos na literatura [Uttamatin et al. 2013b]. Além do método de Otsu, o trabalho foi comparado com a abordagem adaptativa e *fuzzy*, quando aplicadas isoladamente. A Figura 6 mostra os resultados de segmentação para os métodos Otsu, adaptativo, *fuzzy* e *fuzzy*-adaptativo.

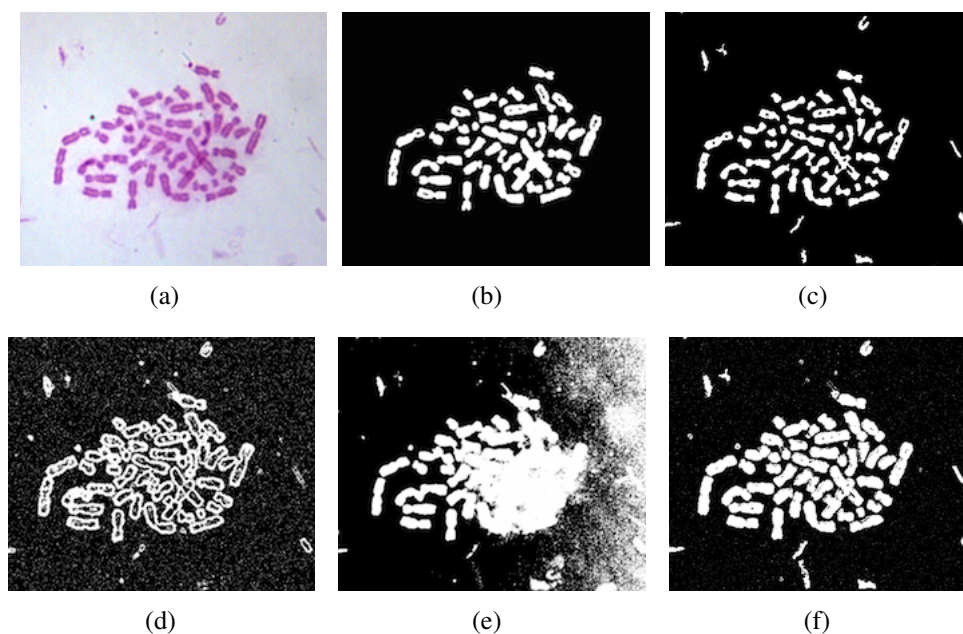


Figura 6. Segmentação de imagem de cromossomos por diferentes técnicas: (a) imagem original, (b) imagem ouro, (c) Adaptativo, (d) *Fuzzy*, (e) Otsu e (f) *Fuzzy-Adaptativo*.

A Figura 6 (a) representa a imagem original enquanto que a coluna (b) mostra a imagem ouro. As Figura 6 (c)-(d) mostram os resultados de binarização para as técnicas adaptativa, *fuzzy*, Otsu e *fuzzy*-adaptativo, respectivamente. É possível observar que o método de Otsu possui dificuldade de segmentação em algumas imagens devido à variação de iluminação. O método adaptativo, apesar de conseguir segmentar bem os cromossomos na maioria dos casos, às vezes acaba eliminando algum componente do cromossomo. Porém, esses componentes conseguem ser preservados na abordagem *fuzzy*-adaptativa.

A Figura 7 mostra os resultados quantitativos da segmentação de cada técnica, para as métricas analisadas. Os valores de parâmetros utilizados em cada técnica foram definidas através da análise de variação de parâmetros descritos anteriormente, onde foram definidos os melhores parâmetros baseados nas métricas estudadas. A partir dos resultados da Figura 7, pode-se observar que o método proposto obteve maiores valores do que as outras técnicas para as métricas de sensibilidade, especificidade, acurácia

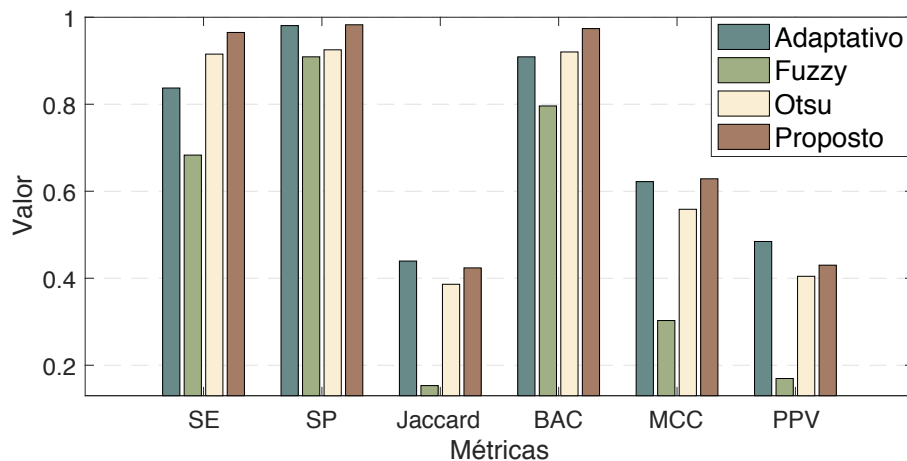


Figura 7. Média das métricas para as imagens segmentadas pelas técnicas de limiarização adaptativa, *fuzzy*, Otsu e método proposto.

balanceada e MCC. Para a métrica de Jaccard e PPV o método adaptativo obteve melhores resultados. No entanto, o método adaptativo obtém baixo valor de sensibilidade comparado às outras técnicas. A sensibilidade é uma métrica importante para assegurar a detecção dos cromossomos, pois ela está relacionada à taxa de verdadeiro positivo. Os resultados também mostraram que o método proposto obteve melhores resultados do que o método de Otsu, que é um método de segmentação bastante utilizado nos trabalhos do estado da arte. A abordagem híbrida desenvolvida mostrou que a composição de técnicas de segmentação pode ser vantajosa para a segmentação de cromossomos.

6. Conclusão

Este trabalho propõe um novo método de segmentação de cromossomos baseado em uma abordagem híbrida de limiarização adaptativa com lógica *fuzzy*. O método desenvolvido é aplicado para 35 imagens da base do CRCN-NE e comparado com técnicas da literatura para as métricas de sensibilidade, especificidade, acurácia balanceada, Jaccard, MCC e PPV.

Resultados mostraram que o método *fuzzy*-adaptativo obtém melhores valores do que métodos bastante usados no estado da arte, como o método de Otsu. Quando comparado com a limiarização adaptativa sozinha, o método proposto consegue obter maior valor de sensibilidade e especificidade, gerando melhores resultados de acurácia balanceada. O trabalho desenvolvido mostra que a abordagem híbrida para segmentação de cromossomos é favorável para a qualidade da segmentação. Além disso, o método contribui para o estado da arte, obtendo melhores resultados do que algoritmos existentes. Trabalhos futuros envolvem a extensão da base de dados e adição de uma etapa de classificação.

Referências

- Arora, T. and Dhir, R. (2016). A review of metaphase chromosome image selection techniques for automatic karyotype generation. *Medical & biological engineering & computing*, 54(8):1147–1157.
- Burger, W. and Burge, M. J. (2016). *Digital image processing: an algorithmic introduction using Java*. Springer.

- Cunha, D. M. d. C. et al. (2015). Análise cromossômica por microarranjos em probandos com indicação clínica de síndrome de down sem alterações cariotípicas.
- Dubois, D. and Prade, H. (1978). Operations on fuzzy numbers. *International Journal of systems science*, 9(6):613–626.
- Dubois, D. J., Prade, H., and Yager, R. R. (2014). *Readings in fuzzy sets for intelligent systems*. Morgan Kaufmann.
- Grisan, E., Poletti, E., and Ruggeri, A. (2009). Automatic segmentation and disentangling of chromosomes in q-band prometaphase images. *IEEE Transactions on Information Technology in Biomedicine*, 13(4):575–581.
- Haykin, S. S. (2001). *Redes neurais*. Bookman.
- Jianzhuang, L., Wenqing, L., and Yupeng, T. (1991). Automatic thresholding of gray-level pictures using two-dimension otsu method. In *Circuits and Systems, 1991. Conference Proceedings, China., 1991 International Conference on*, pages 325–327. IEEE.
- Khmelinskii, A., Ventura, R., and Sanches, J. (2008). Automatic chromosome pairing using mutual information. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 1918–1921. IEEE.
- Kurtz, G. C. (2011). Metodologias para detecção do centrômero no processo de identificação de cromossomos.
- Madian, N., Jayanthi, K., and Suresh, S. (2015). Contour based segmentation of chromosomes in g-band metaphase images. In *Signal and Information Processing (Global-SIP), 2015 IEEE Global Conference on*, pages 943–947. IEEE.
- Matta, M. C. d. (2013). Processamento de imagens em dosimetria citogenética.
- Minaee, S., Fotouhi, M., and Khalaj, B. H. (2014). A geometric approach to fully automatic chromosome segmentation. In *Signal Processing in Medicine and Biology Symposium (SPMB), 2014 IEEE*, pages 1–6. IEEE.
- NUSSBAUM, R. L., MCINNES, R. R., and Willard, H. (2008). Thompson: Genética médica.
- Pantaleão, C. H. Z. et al. (2003). Contribuição à análise e classificação citogenética baseada no processamento digital de imagens e no enfoque lógico-combinatório.
- Poletti, E., Zappelli, F., Ruggeri, A., and Grisan, E. (2012). A review of thresholding strategies applied to human chromosome segmentation. *Computer methods and programs in biomedicine*, 108(2):679–688.
- Uttamatin, R., Intarapanich, A., Kaewkamnerd, S., Yuvapoositanon, P., and Tongsima, S. (2013a). Band classification based on chromosome shapes. In *Biomedical Engineering International Conference (BMEiCON), 2013 6th*, pages 1–5. IEEE.
- Uttamatin, R., Yuvapoositanon, P., Intarapanich, A., Kaewkamnerd, S., Phuksaritanon, R., Assawamakin, A., and Tongsima, S. (2013b). Metasel: a metaphase selection tool using a gaussian-based classification technique. *BMC bioinformatics*, 14(16):S13.

Boosting machine learning techniques for wood quality classification by particle swarm optimization

Mateus Roder¹, Carlos de Oliveira Affonso¹, André Luis Debiaso Rossi¹

¹Universidade Estadual Paulista (UNESP) – Câmpus de Itapeva
CEP: 18409-010 – Itapeva – SP – Brazil

{mateusroder}@hotmail.com, {affonso, alrossi}@itapeva.unesp.br

Abstract. *This paper deals with the problem of automatically selecting the hyperparameters of machine learning techniques for wood quality classification using particle swarm optimization (PSO). The main aim is to analyze the optimization impact in predictive performance of the induced models. The techniques employed to generate these models are Support Vector Machines and Neural Networks, that are most used for this task. Experiments were carried out using wood boards images from a Brazilian sawmill industry. The experimental results show that PSO was able to find hyperparameter values that lead to better predictive performance of both machine learning techniques compared to two baselines. The most significant improvement was observed for the neural network, where PSO reduced the error rate by around 6% compared to default parameter values and by 3% compared to a random solution generator.*

1. Introduction

The inspection and classification of wood boards are tasks of great importance for suppliers of this raw material. These procedures are usually performed by human operators trained for this purpose. However, this solution suffers considerable disadvantages, such as subjectivity, due to fatigue, for example, after performing the same task over an extended period, or due to operators with different expertise and views.

For this reason, it is possible to find in the literature several studies that investigate Machine Learning (ML) techniques for automated visual inspection, or more specifically, wood quality and defects classification [Silvén and Niskanen 2003], [Pham et al. 2006], [Cavalin et al. 2006], [Roder et al. 2015]. ML techniques have achieved great success in performing specific tasks such as data mining, pattern recognition, and inspection and controlling the quality of products and processes [Flach 2012, Koksai et al. 2011]. Machine Learning is a research area that studies how to develop systems capable of gaining knowledge automatically with experience. In this sense, experience can be understood as a set of examples that is used by a learning algorithm to induce models. Generally, the more data (experience) is available, the better the learned models can be. Later, these models are applied to predict future cases or to other types of decision making under uncertainty [Flach 2012].

A computational intelligent system capable of classifying wood quality due to defects, such as cracks, knots and exposed pith, has been developed [Affonso et al. 2015, Vieira 2016]. The authors collected images from wood boards in a wood sawmill company from Brazil, where a specialist selects the wood according to their quality in three different levels (Fig. 1):

- **A:** if the wood has no defects;
- **B:** if the wood has a few defects;
- **C:** if the wood has defects that can harm the product quality.

Although ML had been successfully employed for this problem, these techniques have hyperparameters to adjust their bias and, consequently, influence directly in predictive performance. The search for the best parameter values usually involves the optimization of a large search space, making this problem very costly. However, the most common approach for this task is trial-and-error, which is highly subjective and expensive. Other simple methods, such as exhaustive searching, are intractable when there are more than two parameters to be adjusted [Basheer and Hajmeer 2000, Chapelle et al. 2002].

Hence, metaheuristics based on the evolutionary process and swarm intelligence have been investigated as an alternative to automatically adjust the hyperparameters of ML algorithms [Ren et al. 2014] [Rossi and Carvalho 2008] [Gomes et al. 2012]. Some of these metaheuristics are Genetic Algorithms (GAs), Simulated Annealing (SA) [Castro 2006] and Particle Swarm Optimization (PSO) [Kennedy and Eberhart 1995]. Among them, PSO is one of the most used technique for this purpose and which achieved the best results [Rossi and Carvalho 2008, Rossi et al. 2008, Gomes et al. 2012, Bao et al. 2013].

PSO is inspired in the collective behavior of developed systems and employed for optimization problems that have to get robust solutions. In the last 20 years, this bioinspired algorithm has been widely studied and employed. In addition, some variants have been proposed aiming to obtain improvements for some optimization problems [Bao et al. 2013].

The main aim of this paper is to automatically tune the hyperparameters of neural networks and support vector machines employed to classify wood timber images. As a result, it is expected to improve the predictive performance of the generated models for this task. The PSO is investigated for this purpose and its results is compared with baseline methods, namely default parameter values of libraries which implement the ML techniques and a randomized solution generator.

The remaining of this paper is as follow. In Section 2, we briefly present the most relevant related work. Section 3 presents the machine learning techniques and the particle swarm optimization algorithm investigated in this paper for parameter tuning. The material and methods for the experiments carried out here are presented in Section 4. Next, in Section 5, the experimental results are shown and analyzed. Finally, Section 6 provides an overview and highlights the achievements and the relevance of this paper.



Figure 1. Different wood quality classified by an specialist. Source: [Vieira 2016]

2. Related Work

Several studies have successfully employed PSO for hyperparameter tuning of ML techniques for different problems [Gomes et al. 2012] [Rossi et al. 2008] [Lin et al. 2009] [Bao et al. 2013] [Ren et al. 2014]. The studies described in this section are related to this paper regarding both PSO and the ML techniques, namely Artificial Neural Network [Haykin 1999] and Support Vector Machines [Cristianini and Shawe-Taylor 2000].

The architecture (input and hidden nodes and connections) and the weights of a Multilayer Perceptron (MLP) neural network were optimized by a PSO algorithm in [Ludermir and de Oliveira 2013]. In this study, they employed the network for the identification of factors related to common mental disorders. The experimental results showed that the optimization was an interesting alternative to the statistic model of logistic regression for this task. The main hypothesis is that it was possible because the neural network was able to detect all the possible interactions among the many explaining variables.

A hybrid PSO was used by [Ren et al. 2014] to select optimal parameters of a backpropagation ANN for wind speed forecast. In that work, the authors used PSO to select which parameters of a MLP resulted in better predictive performance for the wind speed problem. The MLP optimized by the PSO algorithm outperformed other regression methods, such as ARIMA and a MLP without parameter optimization.

Another study of MLP parameters tuning was presented in [Rossi et al. 2008]. The authors studied whether bioinspired algorithms, such as ant colony optimization, genetic algorithm and PSO, would be able to improve the predictive performance of MLP networks for gene expression analysis by optimizing MLP hyperparameters. Experimental results over four gene expression data sets showed that ACO lead to the smallest error rates for two data sets. On the other hand, PSO was very competitive for all problems under analysis.

The work of [Gomes et al. 2012] proposed the combination of meta-learning and search algorithms to deal with the problem of SVM parameter selection. The hyperparameter values obtained by meta-learning (kernel and regularization parameters) were used as initial search points for PSO. Overall, PSO combined with the meta-learning refined its initial solutions, which were closer to the best results found.

In the study of [Bao et al. 2013], PSO and pattern search (PS), named PSO-PS based memetic algorithm (MA), was proposed and used to optimize the parameters of SVMs (kernel and regularization parameters). In the proposed method, PSO is responsible for exploration of the search space and the detection of the potential regions with optimum solution while PS is used to produce an effective exploitation on the potential regions obtained by PSO. The novel method achieved better results than other techniques.

3. Machine learning techniques and PSO

The task of wood quality classification is addressed in this paper by two ML techniques: Artificial Neural Network (ANN) and Support Vector Machines (SVM). As previously mentioned, besides their high predictive accuracy for this problem, they have hyperparameters which must be defined by users. In this paper we analyze the potential improvement of using PSO for this purpose. Thus, in this section, we briefly present these ML

techniques and the main PSO concepts.

3.1. Multi-layer Perceptron Neural Network

The most popular ANN architecture is the Multi-Layer Perceptron (MLP), which has been successfully employed in a wide variety of problems [Haykin 1999]. MLP networks have one or more hidden layers of neurons whose connection weights are usually adjusted by gradient-based learning algorithms. These networks can be fully or partially connected. A neural network is called fully connected when each node of a layer is connected to all nodes of the next layer and is named partially connected when some neurons of adjacent layers are not connected [Haykin 1999].

Besides the number of hidden layers and neurons, other parameters influence MLP predictive performance. The learning rate (η) determines the speed of learning. A very low rate makes learning very slow, while a very high rate causes oscillations in training and prevents the convergence of the learning process. Generally, its value ranges from 0.1 to 1.0. Moreover, a MLP has the term momentum (μ) accountable to increase the speed of neural network training and reduce the instability or avoid local minima. This term may or may not be used during training and its value ranges from 0 (non use) to 1.0 [Haykin 1999].

3.2. Support Vector Machines

Recently the SVMs have attracted the attention of the machine learning community due to their good performance, sometimes equal to or superior to that achieved by other techniques, such as ANN [Cristianini and Shawe-Taylor 2000].

The SVMs are based on the theory of structural risk minimization [Cristianini and Shawe-Taylor 2000]. It searches an optimal hyperplane as a decision surface that is able to separate the examples with maximum distance. In this context, if the data of a problem are linearly separable, SVMs are named linear with a linear decision boundary. In the other hand, non-linear data need non-linear SVMs to deal with the problem. Non-linear SVMs with smooth margins allow to handle outliers and noise by introducing slack variables to the constraint of the primal optimization problem, making it possible to smooth the margins of the classifier. In this context, the high dimensional feature space is necessary for trying to transform the non-linear data into linear. This transformation becomes important to simplify the problem when the data have a high dimension feature [Haykin 1999].

3.3. Particle Swarm Optimization

The PSO is a technique inspired in the collective behavior of developed systems and implemented in optimization problems that have to get robust solutions. For the PSO, the collective represent the swarm.

The particles move through a search space and are able to store past information and share information with other particles. These two types of information correspond to individual (cognitive) learning and cultural (social) transmission. In this way, particles use the best solutions in their evolutionary process.

Moreover, each individual uses three principles in the process of cultural adaptation [Castro 2006]:

- **Evaluate:** Individuals are able to sense the environment to estimate their behavior.
- **Compare:** Individuals compare to baseline.
- **Imitate:** A central part of human social organizations and is important for acquiring and maintaining mental abilities.

Information sharing is performed between particles considered to be neighbors. The neighborhood considered by the PSO is topological and does not refer to the attribute space of each particle. There are basically two ways of defining neighborhood in this context: global and local. The global considers that each particle is connected to all the others and so sharing of information occurs between the whole swarm. In the local neighborhood each particle is composed of itself and its closest k -neighbors [Castro 2006].

The PSO can be applied basically to binary or continuous problems. Each particle is treated as a point in a D -dimensional space and represented by its current position, velocity and the best position found. The position of the particle i is given by $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$; Its velocity is given by $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ and the best position found by this particle is given by $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ [Castro 2006]. In the case of the global neighborhood, the best position found among all particles is represented by p_g .

A particle will move in a certain direction depending on its current position, its speed, the best position found by it and the best position found by its neighbors. Equations 1 and 2 determine how the velocity and position of the particles are updated, respectively. In order to avoid the explosion of particle velocity, a speed limit has been introduced for the limits maximum and minimum (v_{max} and $-v_{max}$ respectively). The position of the particle may also be restricted to the defined search space interval.

$$v_{id}(t+1) = \omega \cdot v_{id}(t) + \varphi_1 \cdot r_1 \cdot (p_{id} - x_{id}(t)) + \varphi_2 \cdot r_2 \cdot (p_{gd} - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t) \quad (2)$$

where t represent the iteration, ω the inertia weight for balance the global and local search proposed by [Shi and Eberhart 1998], r_1 and r_2 are two independent values with uniform distribution between $[0; 1]$, φ_1 and φ_2 are acceleration constants.

4. Experiments

This section presents the experiments carried out to evaluate the PSO algorithm in order to tune the hyperparameters of both MLP and SVM, which are employed for the problem of wood quality classification. Next, the image data set collected for this study, the hyperparameters of the ML techniques and the methodology to optimize them through PSO are presented.

4.1. Data

The image data set used by the ML techniques to generate models for the classification of wood quality contains 374 instances (images), which were obtained from the analysis of wood boards in a Brazilian sawmill industry [Vieira 2016]. Each instance is composed of the values of the predictive attributes and the target attribute. The predictive attribute values were obtained through the extraction of characteristic from the wood boards images

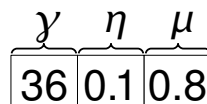


Figure 2. Representation of an individual for a network MLP.

using the measures proposed by [Haralick et al. 1973], namely entropy, energy, correlation, maximal correlation coefficient and inverse difference moment. For each measure the nearest neighbor in relation to 0° and 45° angular dependencies were computed, what generated an amount of 10 predictive attributes. The target attribute identifies the quality of the boards according to a standard proposed by the industry experts. The quality “A” defines a high quality standard, “B” an intermediate quality and “C” a lower quality. From the amount of 374 instances generated, 144 are of class “A”, 177 are of class “B” and 53 of class “C”. Therefore, there is an unbalance of classes in this data set, because, in practice, this is the proportion number of instance for each class.

4.2. Machine Learning Techniques

As previously mentioned, wood quality classification was performed by two ML techniques in this paper, namely MLP and SVM. The MLP was trained by the backpropagation algorithm with momentum. In the experiments, only one hidden layer was used, which allows the representation of numerous functions, sufficient for the purpose of this study. In addition, this network is fully connected. The parameters adjusted by the PSO are the number of neurons in the hidden layer (γ), the learning rate (η) and the term momentum (μ). In these experiments, we used the Weka [Hall et al. 2009] though the R Language and Environment for Statistical Computing [R Core Team 2015].

For SVMs was used the e1071 package [Meyer et al. 2015], which is an interface to the LIBSVM library. This library implements different kernels for the SVMs, and each of them has different parameters. The RBF (Gaussian) kernel will be used in this research because it usually obtains the best results and has fewer numerical difficulties than other kernels [Hsu et al. 2007]. In this case, it is necessary to set two parameters, one referring to the RBF kernel, named gamma (γ) and the other one referring to the SVM itself, named cost (C).

4.3. PSO for parameter tuning

In the optimization process, each parameter related to the ML techniques is represented directly as a particle, which is the candidate solution for the problem. An example of this representation is shown in Figure 2, for an MLP network, with the three parameters investigated in the study. The stopping criteria of PSO can be, for example, a maximum number of iterations or a minimum error achieved. In this study, the first approach was used, since the other one may underestimate or overestimate the minimum error.

The PSO parameters were set as described in Table 1. The parameter v_{max} is inherent to each technique, and its values were established for a range of each ML parameter. For the MLP, v_{max} of $\gamma = [1, 18]$, $\eta = [0, 1]$ and $\mu = [0, 1]$. For the SVM, v_{max} of $C = [0, 10]$ and $\gamma = [0, 4]$. An amount of 15 replications of the algorithm using these settings were carried out to obtain a more reliable estimate of the PSO and ML performance, since PSO is a stochastic population-based search algorithm.

Table 1. PSO parameter values.

Parameter Description	Symbol	Value
Number of particles	P	30
Acceleration Constant 1	C_1	1.494
Acceleration Constant 2	C_2	1.494
Inertial Weight	ω	0.729
Maximal Velocity	v	v_{max}
Iterations	i	500

The PSO algorithm was implemented in R language. Next, it was evaluated for five benchmark functions, as in [Elshamy et al. 2007], namely Sphere, Rosenbrock, Rastrigin, Schaffer f6 and Ackely. These functions are interesting to test the ability of the algorithm to “escape” from local minima. Sphere and Rosenbrock functions are unimodal whereas the others are multimodal, with a large number of local minima. However, they all have a global minimum of zero. For each function, ten replications were performed, that is, the seed of generation of the initial population and the values to be updated of the particles ranged from 1 to 10. In this evaluation, PSO found values very close to the global minimum for almost all functions, confirming it could be applied for the purpose of parameter tuning.

4.4. Baselines

Libraries and packages of ML techniques usually define default hyperparameters values for their code. In this study, the predictive performance of models induced with these values are used as baseline, namely Default, in this study. For the MLP network, the number of neurons in the hidden layer (γ), the learning rate (η) and the term momentum (μ) are set by Weka as 2, 0.3 and 0.2, respectively. For SVMs, the kernel parameter gamma (γ) and cost (C) are set by its package as 0.01 and 1.0, respectively. Thus, ML techniques were used to induce a classifier model, posteriorly optimized by the PSO. These default parameters values of both techniques were also analyzed in [Roder et al. 2015].

In addition to the comparison with Default values, a random solution generator (RSG) was employed. The RSG operates based on the generation of random solutions considering the search space limits, but without using any PSO mechanism. In order to be as fair as possible, the same number of solutions tested by PSO are randomly generated at each iteration of RSG over the same search space. Thus, an amount of 30 solutions are randomly generated through 500 iterations (Table 1). The “fitness” of these solutions are assessed using the ML techniques, i.e., an amount of 15 000 solutions.

4.5. Performance assessment

In the experimental method proposed in this article, the generation and evaluation of the classification models will be based on the stratified k -fold cross-validation. In this method, one of the k partitions is separated for testing and another is separated for validation, i.e., estimates of the generalization power of the models. The remaining $k - 2$ partitions are used for model training. The average error obtained for the k validation partitions is the average estimate for the test error of the classifiers. This estimation is used by PSO to guide its search in the space of solutions in search of the optimal set of values

Table 2. Predictive performance of MLP models assessed by error rate and kappa for the test set.

MLP	Error Rate	Kappa
PSO	0.130 ± 0.005	0.781 ± 0.009
Default	0.187 ± 0.000	0.691 ± 0.000
RSG	0.156 ± 0.000	0.739 ± 0.000

Table 3. Predictive performance of SVM models assessed by error rate and kappa for the test set.

SVM	Error Rate	Kappa
PSO	0.147 ± 0.003	0.746 ± 0.005
Default	0.190 ± 0.000	0.645 ± 0.000
RSG	0.153 ± 0.000	0.742 ± 0.000

for the hyperparameters of the ML techniques. Thus, for each solution (combination of parameter values), the learning algorithm is computed k times.

Besides the error rate, the Cohen's Kappa was used to assess the predictive performance of the models because it considers the class distribution of the data set. It represents (approximately) the correlation between the scores of two raters. The value variate between -1.0 and +1.0, which -1.0 represents complete disagreement between raters and +1.0 complete agreement [Kvålseth 1989].

5. Experimental Results

This section presents the experimental results of tuning the hyperparameter values of both MLP network and SVM techniques for the task of wood quality classification. For such, a PSO algorithm was designed for this purpose. The predictive performance of the induced models are compared with baseline methods.

The error rate and kappa values achieved by MLP network for hyperparameter tuning are show in Table 2. These values are the average of the 15 replications of PSO and RSG. Analyzing this table, one can see that PSO was able to reduce the error rate of MLP models by approximately 6% and 3% compared with Default and RSG, respectively. Although the error rate has been used during the tuning process (fitness), the kappa values obtained by the PSO were greater than Default and RSG (9% and 4%, respectively). Thus, we can observed that both performance measures keep some relation, despite the problem under analysis is classes unbalanced. Thus, the kappa results are interesting by the fact that PSO searches the global minimum for the fitness function using the error rate measure.

The same analysis was performed for the SVM models (Table 3). Similarly to the other ML technique, the PSO lead SVM to generate models with smaller error rates compared to Default and RSG, approximately 4% and 0.6%, respectively. Moreover, it is also possible to observe an improvement regarding kappa values of 10% and 0.4% compared to Default and RSG, respectively. These results indicate a superiority of PSO for the task of hyperparameter tuning of these ML techniques applied to wood quality classification.

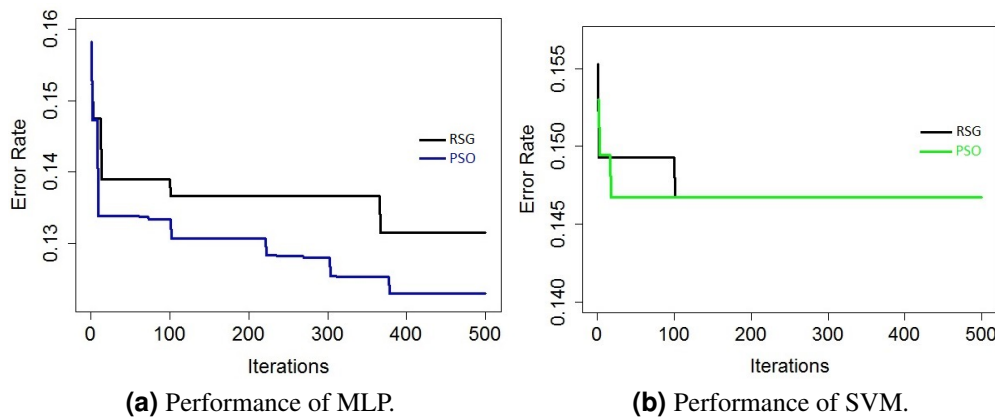


Figure 3. Performance of ML techniques.

In order to obtain more evidence about the significance of the differences among PSO and baseline methods, the Wilcoxon's test was applied. This test used the predictive performance of ML techniques (error rate and the kappa) when their hyperparameters were defined by PSO, RSG and Default. For such, we considered the 15 replications of the experiments.

Comparing PSO and Default, the p-values for both SVM and MLP were smaller than 0.01 considering any performance measure. These results show that there is a significant difference between the methods for both techniques. The comparison between PSO and RSG was similar to the previous. Once the p-value were again smaller than 0.01 for both ML techniques, the improvement in performance achieved by PSO was significant compared with RSG.

In order to assess the evolution of the parameter values found by PSO and RGS over iterations, the predictive performance of ML techniques for the 12th replication were plotted (Figure 3). The fitness values obtained by RSG, were organized in order to visualize how would be the evolution in case the algorithm had some mechanism to keep the best solution during the generations. In other words, during the iterations the minimum value was kept as "global minimum" in the plot.

For the MLP, we have to observe two main characteristics in Figure 3a. First, the error rate obtained by PSO was patent superior to RSG. Second, this graphic shows that PSO got stuck in some local minima for some iterations, but it was able to scape from them faster than RSG. Consequently, PSO found smaller error rates after the same number of iterations.

From Figure 3b, one can note that PSO quickly reached its best solution (18 iterations, i.e., 540 solutions), with posterior stabilization in a local minimum. On the other hand, RSG found the same results after more than 100 iterations, i.e., after analyzing more than 3000 solutions.

6. Conclusion

In this papper we analyzed the problem of tuning free parameters related to ML techniques (MLPs and SVMs) aiming to improve their predictive performance for wood quality classification using PSO. The parameters of MLPs and SVMs are of great importance

to the techniques success. Experimental results showed that PSO was able to find hyperparameter values which lead the ML algorithms (MLP and SVMs) to gain in classification accuracy. These experiments also provided insights that PSO performs better than baseline methods for the same purpose.

In addition to the predictive performance improvement, the PSO converged more quickly to the best search space than RSG. While PSO took only 18 iterations for the SVM, RSG found the same result after more than 100 iterations. Regarding MLP, these numbers are greater for both PSO and RSG. They found their best values after more than 350 iterations. However, the reduction obtained by PSO was significantly better, with approximately 6% greater than Default and 3% greater than RSG. In addition, we also observed a significant increasing in both techniques for the cohen's kappa measure. This is noteworthy because there was an agreement between the two measures used for performance evaluation, although the data set is class unbalance and only the error rate was used as fitness function.

Overall, this paper confirms the necessity of parameters adjustment of ML techniques applied for the problem of wood quality classification. Moreover, experimental results showed the efficiency of PSO for task, even if in some cases the RSG achieved similar values, its convergence was slower than PSO. As a future work, we plan to study other parameters to be adjusted (i.e., the number of hidden layers) and other ML techniques for the wood quality classification, such as Convolutional Neural Networks.

References

- Affonso, C., Sassi, R. J., and Barreiros, R. M. (2015). Biological image classification using rough-fuzzy artificial neural network. *Expert Systems with Applications*, pages 9482–9488.
- Bao, Y., Hu, Z., and Xiong, T. (2013). A pso and pattern search based memetic algorithm for svms parameters optimization. *Neurocomputing*, 117:98–106.
- Basheer, I. and Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43(1):3–31. Neural Computing in Micrbiology.
- Castro, L. N. (2006). *Fundamentals of natural computing: Basic concepts, algorithms, and aplications*. Chapman & Hall/CRC.
- Cavalin, P., Oliveira, L. S., Koerich, A. L., and Britto, A. S. (2006). Wood defect detection using grayscale images and an optimized feature set. In *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, pages 3408–3412.
- Chapelle, O., Vapnik, V., Bousquet, O., and Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159.
- Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support Vector Machines: and Other Kernel-Based Learning Methods*. Cambridge University Press, New York, NY, USA.
- Elshamy, W., Emara, H. M., and Bahgat, A. (2007). Clubs-based particle swarm optimization. In *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*, pages 289–296. IEEE Computer Society.

- Flach, P. (2012). *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, New York, NY, USA.
- Gomes, T. A., Prudêncio, R. B., Soares, C., Rossi, A. L., and Carvalho, A. (2012). Combining meta-learning and search techniques to select parameters for support vector machines. *Neurocomputing*, 75(1):3–13.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18.
- Haralick, R., Shanmugam, K., and Dinstein, I. (1973). Textual features for image classification. In *Proceedings of the IEEE Transaction on Systems, Man and Cybernetics*, pages 610–621. IEEE.
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.
- Hsu, C. W., Chang, C. C., and Lin, C. J. (2007). *A practical guide to support vector classification*. Department of Computer Science - National Taiwan University.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948.
- Koksal, G., Batmaz, I., and Testik, M. C. (2011). A review of data mining applications for quality improvement in manufacturing industry. *Expert Systems with Applications*, 38(10).
- Kvålseth, T. O. (1989). Note on cohen’s kappa. *Psychological Reports*, 65(1):223–226.
- Lin, S.-W., Chen, S.-C., Wu, W.-J., and Chen, C.-H. (2009). Parameter determination and feature selection for back-propagation network by particle swarm optimization. *Knowledge and Information Systems*, 21(2):249–266.
- Ludermir, T. B. and de Oliveira, W. R. (2013). Particle swarm optimization of mlp for the identification of factors related to common mental disorders. *Expert Systems with Applications*, 40(11):4648–4652.
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2015). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.6-7.
- Pham, D. T., Soroka, A. J., Ghanbarzadeh, A., Koc, E., Otri, S., and Packianather, M. (2006). Optimising neural networks for identification of wood defects using the bees algorithm. In *2006 4th IEEE International Conference on Industrial Informatics*, pages 1346–1351.
- R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Ren, C., An, N., Wang, J., Li, L., Hu, B., and Shang, D. (2014). Optimal parameters selection for {BP} neural network based on particle swarm optimization: A case study of wind speed forecasting. *Knowledge-Based Systems*, 56:226–239.
- Roder, M., Rossi, A. L. D., and Affonso, C. O. (2015). Comparação de técnicas de aprendizado de máquina para a classificação da qualidade da madeira. 8^o Congresso de extensão universitária da UNESP, pages 1–6.

- Rossi, A. L. D. and Carvalho, A. C. P. L. F. (2008). Bio-inspired optimization techniques for svm parameter tuning. In *Proceedings of 10th Brazilian Symposium on Neural Networks*, pages 435–440. IEEE Computer Society.
- Rossi, A. L. D., Carvalho, A. C. P. L. F., and Soares, C. (2008). Bio-inspired parameter tuning of MLP networks for gene expression analysis. In *Proceedings of 8th International Conference on Hybrid Intelligent Systems*, pages 57–62. IEEE Computer Society.
- Shi, Y. and Eberhart, R. (1998). A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 69–73.
- Silvén, O. and Niskanen, M. (2003). Framework for industrial visual surface inspections. In *Quality Control by Artificial Vision*, pages 1–6. International Society for Optics and Photonics.
- Vieira, F. H. A. (2016). *Image processing through machine learning for wood quality classification*. PhD thesis, Faculdade de Engenharia de Guaratinguetã (FEG), UNESP.

Using Intelligent Agents to Optimize an Investment Portfolio

Lucas O. Souza¹, Celia G. Ralha¹

¹ Computer Science Department
University of Brasilia (UnB)
Brasilia, DF – Brazil

lucasosouza@gmail.com, ghedini@unb.br

Abstract. *Forecasting of stock prices, a key component in the asset allocation policy of investment funds, is generally modeled as a regression problem. However, regression often leads to unreliable results, contributing to the belief that future prices in an open market are unpredictable. This paper presents a novel approach to the investment fund problem, by modeling the forecast as a set of binary classifiers which predicts whether predefined strategies can be successful. The model serves as the main engine of a multiagent system that leverages the predictions to define an optimal resource allocation policy. Results achieved are promising and point to a new direction of research in asset allocation in investment portfolios.*

1. Introduction

Choosing to buy assets based on a predictive upward trend is a discipline that dates back to the early days of capitalism. In 1637, in Holland, there was a non expected surge in tulips prices due to speculation, that caused a tulip bubble which today is widely regarded as the first stock market crash [1]. The bubble was caused by investors buying tulips with the expectation that its price would rise in the long term.

These events are more frequently in the last 30 years due to the introduction of computing intelligence in the stock trading game. Technical analysts have tried to predict future trends solely based on price and volume historical data, using esoteric methodologies of charts interpretation that even includes Leonardo da Vinci's infamous golden ratio [2]. Fundamental analysts have manipulated millions of spreadsheets to come upon the perfect expected price of the company and based on it predict whether its price would rise or fall.

In 2013, Eugene Fama won the Nobel Prize for Efficient Market Hypothesis, which states that all relevant information regarding a company is already contained in its stock price, having no possibility therefore for prediction of stock price trends [3]. That assumption has been challenged for many decades, but it has gained new colors with the rise of data science and machine learning as research field.

The introduction of artificial intelligence (AI) methodologies in investment portfolio management and the rise of machine learning have drawn a lot of interest, which promises big rewards for those who can crack the challenge. When a bank or a hedge fund decides which stock to buy and which to sell, many players are involved in the decision. There are those who gather the data, those who process it and try to come up with stock price trends, and those who use this information to trade. The described players'

rules can be well represented by multiagent systems that resembles a pattern encountered in real day-to-day environments [4]. In this paper we discuss a solution that follows this recipe [5], named ORACLIA, short for **Optimizing Resource Allocation and Capital Lucrativeness with Intelligent Agents**.

The rest of the paper includes: in Section 2, a brief explanation of the key concepts approached in the work; in Section 3 we cover the proposed solution, a multiagent system that optimizes portfolio return; in Section 4 we discuss the learning methodology applied; in Section 5 we cover previous research in the related fields; in Section 6 the experimental results are presented; and finally, in Section 7, conclusions and future work are discussed.

2. Background

This paper builds on research from finance and artificial intelligence domains. The most relevant concepts to our solution are introduced below.

2.1. Stock market and trading strategies

Stock market is an open market where players buy or sell assets or stocks, which is the nomenclature used herefore (contracts is another common term found in the literature). An asset is not necessarily of physical nature; it may represent a small share of a company, gallons of oil, silos of grains, or even abstract concepts such as “the opportunity to buy or sell asset XY at a set price 30 days in the future”, a type of derivative.

Bovespa is the most relevant stock market in Brazil. Ibovespa is an index composed of 100 assets most traded in Bovespa (not including derivatives), weighted by their trade volume, and is often used as a benchmark to compare an investment fund performance.

Investment funds operate to maximize the return of investment (ROI) on a portfolio, subject to constraints defined in the contract between the fund manager and their investors. These restraints are commonly related to the type of assets traded, entry and exit condition, and level of risk exposure. The capital of a fund is the sum of the capital allocated by each of its investors.

A trading strategy is an operation executed by the fund manager in order to maximize its ROI. The most common strategy we can think of is buy an asset, expect its price to increase, and sell it with profit. In a more realistic scenario, though, a trading strategy involves several other aspects. These aspects are usually related to maximize profit and minimize risks.

Complex strategies can be devised, involving several levels of derivative. At large investment funds these strategies are devised automatically by expert systems, and its complexity is often beyond comprehension by a human operator. Capital asset pricing model (CAPM) is a model that represents the relationship between risk and expected return of assets [6], and is widely used in funding management strategies.

2.2. Market analysis

There are two main techniques used by financial analysts to get insights into a stock price direction. The first, fundamental analysis, uses market related data such as interest rate, inflation index, jobs report and international reserves to infer the chances of a company

profiting and its stock prices raise [7]. There are many levels of variables, from general market data, to sector specific indicators (such as energy price for the energy sector), to company specific indicators such as the company's expected profit and loss statement for next quarter.

A second technique looks at the problem at a completely different angle. Technical analysts, or chartists, consider the stock market as a game played by agents, and the stock prices fluctuation can be predicted by understanding the emerging behaviour of the interaction between these agents [8]. The stock price plot is used as a base for analysis. Specific patterns in a plot can indicate an incoming change in market's behaviour. For example, if the assets reaches a price which has never been reached before, this might cause a wave of optimism among investors and quickly raise the asset price. A chartist then takes advantage of these signs to predict stock price trend.

A common technical called bollinger bands represented in Figure 1. Each bollinger band represents the stock price plus or less 2x the standard deviation of a moving average, representing them a range of fluctuation around current price. When the lower bollinger band increases and touches the moving average, it is considered as a sign of overoptimism, and the stock price will decrease. If the opposite happen, the upper bollinger band touches the moving average, it is a sign over pessimism and stock price may increase again.

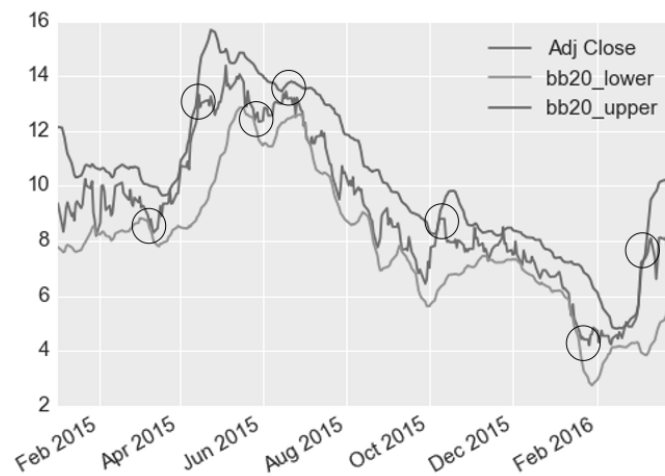


Figure 1. Example of 20 days bollinger bands and trading signs generated by technical analysis

Although technical and fundamental analysis approaches are usually regarded as exclusive, when dealing with it as a machine learning problem we can consider both as useful without lost of consistency. A good model will attribute a high weight to the features which are relevant to prediction, and a lower weight to the remaining, so we can avoid the decision of choosing between the two approaches and use all possible features that could bring some insight [9].

2.3. Supervised learning and cross-validation

As a subset of the large field of machine learning, supervised learning comprises the set of models which can learn a pattern based on pre-existing labeled data. These models

are divided in classification and regression models, with the main distinction being their output [10].

Regression outputs a continuous variable. In the context of stock market, a regression based approach is used to predict an asset price one or more days in the future. Classification, on the other hand, outputs a discrete variable. The number of classes is the number of different values allowed in the output. A 2-class problem is also called binary classification, predicting for example whether the asset price will or will not increase in the following day.

At this work, we use three classifier model of supervised learning: Random Forest, Gradient Boosting and k-Nearest-Neighbors (kNN). Random Forest and Gradient boosting are non-parametric models considered state of the art in supervised learning, while kNN is a non-parametric model widely used in stock price prediction and the method applied in the first version of ORACLIA.

Random Forest is an ensemble method that combines the results of several decision trees through a method called “bagging” [11]. A random forest model selects random samples with replacement and fits a decision tree to each sample. The results are the average prediction of all individual decision trees.

Like random forests, gradient boosting is also an ensemble of weak learners, like decision trees, but uses a boosting strategy instead of bagging [12]. In boosting, each weak learner is trained on the areas of misclassification by previous learners, “boosting” the importance of data points which have not yet been correctly predicted.

The last algorithm, kNN, consists of classifying a new observation by averaging the nearest observations in the feature space [13]. It has a higher space cost since the original training data needs to be preserved, but it trains faster compared to the previously discussed models.

Cross-validation consists of dividing the training set in n folds, and validating the model multiple times, at each time using one fold for testing and $n-1$ folds for training. The results are averaged over all iterations, with the mean being the validation score and the standard deviation a measure of variance that indicates the stability of the model across multiple testing sets.

2.4. Multiagent systems

Multiagent systems (MAS) are systems composed by autonomous agents, which coexists in the same environment, and interact with one another via cooperation or competition. Each agent has its goals and policy, which defines how the agent will act given each environment state.

Communication between agents is done by standard protocols such as defined in Foundation for Intelligent Physical Agents (FIPA)-Agent Communication Language (ACL) [14].

The autonomy of each agent ensures a scalable and flexible system which only relies on the common standard of communication between the agents, allowing for a diverse set of human and computer agents to contribute to the final goal.

3. Proposed Solution

The solution is a two pass vertically oriented multiagent architecture, depicted in Figure 2. The goal is to maximize the return of the portfolio, and one of the relevant subgoals is to increase precision in predicting whether or not a predefined strategy will be successful.

The architecture has three layers, two of which communicate with the exterior world. The portfolio optimization layer receives input from the user of which assets to consider, how much capital to allocate and a time frame, and outputs buy and sell orders. The data gathering layer also communicates with the world to collect data available online, using html scrappers and relevant APIs.

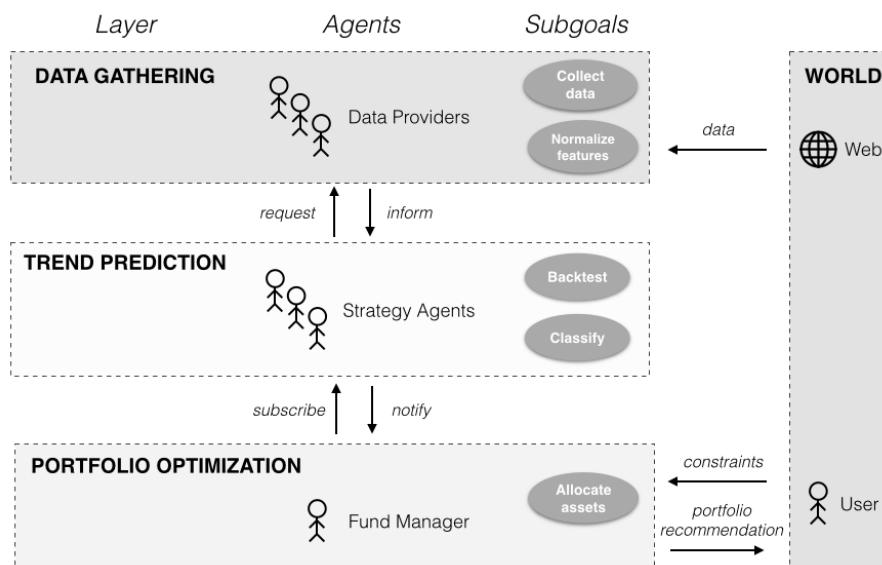


Figure 2. Architecture of the multiagent system ORACLIA

Follow a description of each agent that composes the system.

3.1. Data Providers

Act upon receiving a request for the strategy agent to provide a dataset. The provider is only aware of the intricacies related to the dataset it needs to analyze. For example, a data provider responsible for Twitter data has the required knowledge to gather Twitter data, mine and process it with a sentiment analysis algorithm labeling as positive and negative [15]. The output is the relevant information which is provided to the Strategy Agent via request-inform protocol.

The data gathered and preprocessed is made available to the agents in the second layer, through a request-inform protocol. The strategy agent can subscribe to a service published by the data provider, and receive information when new data is available to be requested.

3.2. Strategy Agents

Play a similar role of data analysts in investment funds. They are responsible for learning the patterns of data, and using this knowledge to assess if a predetermined strategy will or will not work at a given point in time.

The agents have no interaction with the outside world. Strategy agents publish its results, as a yes/no indication of whether or not to follow a given strategy. The fund manager agent will use it as an input to evaluate its optimal course of action.

3.3. Fund Manager

Subscribes to receive strategy recommendations from several strategy agents (in which the simplest one is to buy or sell an asset). May choose whether or not to follow the recommendation and how much capital to allocate, given its available funds and risk appetite.

Its available actions, then, are to choose and apply one of the strategies in a share of its portfolio, balancing risk and maximizing return. The output to the user is a report of the actions taken.

3.4. Software and Hardware

ORACLIA is implemented using several open source Python packages freely available on the Web. Scikit Learn [16] was used for machine learning algorithms and data preprocessing features.

The agents implementation were done with the assistance of Smart Python Agents Development Environment (SPADE) [17]. SPADE is a FIPA-compliance framework to MAS development based on Java Agent Development Framework (JADE).

The infrastructure used was a local MacBook Pro mid-2014 notebook, with 8 cores and 16GB RAM.

4. Model Description

A crucial component of ORACLIA is learning, which we discuss in detail in this section. Learning comprises the three layers presented above. Data providers convert data into features, which are used by strategy agents to train a machine learning model which goal is to predict the success or failure of a trading strategy. The output of the learning model is the input for the fund manager policy on how to optimize return on the portfolio.

4.1. Constructing features

Having selected the variables, we face the problem of turning variables into features for a machine learning model, task executed by data providers. Stock price is a time series data, in which the past influences the future. What matters to prediction then is not the raw numbers, but rather how the variables vary in time. Hence, to identify a trend in stock price, we also need to use the trend of the remaining variables as features for the machine learning model.

To address this, we first define a number of days we will be looking into the past, which can vary according to the time period of the objective we set. For a 10 days time period, the variables for the given day up to 60 days in the past is included in the model. Each pair {variable, day} becomes a feature in the dataset. For 10 variables, the total number of features is $60 \times 10 = 600$.

The next step is to convert the features from raw numbers to a ratio that shows the variable trend in the specified period. Let T be the number of days we will look into the

past, the base date d can be defined as $d - T$. For each variable i , day j , we divide its value by the value of the base date as presented in Equation 1.

$$\Delta X_{ij} = X_{ij}/X_{id} \quad (1)$$

This transformation converts raw values into a delta representing the variable's trend. As an example, if the trading volume in day 30 is 120, and in day 1 is 100, the delta for trading volume is defined as 1.2. Similar transformation is applied to all features in the dataset.

4.2. Defining the labels

With the features created, we turn to labeling, which is the key component in our learning model. We will label the problem as a binary classifier, which outputs 1 or 0.

To label the dataset, we first pick a strategy and an asset in which to apply the strategy. These two variables are defined in the instantiation of a strategy agent. For each observation in the dataset, we will backtest and conclude whether or not the strategy is successful when applied on these conditions. If it is, we label it as 1, and if it does not work, we labeled as 0.

After labeling, we end up with a dataset that relates pattern of features to success or failure of a trading strategy.

4.3. Evaluating performance

The classifiers are evaluated using cross validation on stratified splits. The metric of choice is precision, given by the rate of true positives over positives, meaning how many of the predicted cases are actually positive.

We argue that it is more relevant to avoid incorrect predictions of success (false positives), than to avoid classifying success cases as failures (false negatives). The reason is prediction of success triggers an action by the fund manager, which will incur in loss if incorrect.

Meanwhile, incorrectly classifying a success case as a failure is just a missing opportunity. These can be mitigated by increasing the number of assets under analysis, which increases the overall number of positive cases predicted in the system.

4.4. Trading

Success cases are reported to the fund manager, which use this information to trade. Along with a positive sign, the strategy agents informs the fund manager the likelihood its prediction is true (precision), the expected profit and time span of the trading strategy.

Based on this, the fund manager task is to decide which of the recommended strategies yields the maximum return on its investment, and execute the strategy given its constraints on time, capital and risk appetite.

5. Related Work

In [18] we find a multiagent systems for stock price prediction. The architecture presented by the authors is a four-layered architecture, in which the first layer collects the data

from various sources, both qualitative and quantitative. The second layer preprocess the data, the third layer predict the prices using a bat neural network, and a fourth layer that generates scenarios based on the prediction generated and present report to the decision makers. Only the middle two layers were implemented, with the remaining cited as future work.

Similar approaches are found in previous works [19–22]. While [19] is focused on on predicting stock prices trend, [20] is concerned with investment decisions that optimize asset allocation. In [19], we find multiagent system that uses reinforcement learning for the trading agent, along with traditional strategies for predicting stock price. Author argues that given enough time and data a non-parametric machine learning method can discover complex non-linear relationship that makes the market predictable, and invalidate the Efficient Market Hypothesis. The system is based on previous works on applying Q-learning, recurrent reinforcement learning and adaptive reinforcement learning to stock trading [23–25]. The work [21], presents multiagent system COAST, composed of coaches and advisors, which cooperate to optimize a portfolio based on an analysis of assets risk. In this work the CAPM theory based on market equilibrium concept is applied by autonomous agents to optimize allocation of assets.

Significant work have been done in the field of agents operating in a simulated stock market environment. The Penn Exchange Simulator is a popular simulated environment created as a competition to evaluate autonomous agents in stock market [26]. In [27], the author implements three different agents competing at PXS, a reinforcement learning, a market making, and a trend following agent. A more recent work on intelligent agents in simulated market model [28] also compares three different strategies implemented by independent autonomous agents. Two are fundamental analysis strategies, an averagely informed trader and an insider, and one is technical analysis strategy.

There is a vast literature on the topic of using non-linear regressors to predict stock prices. There seems to be a concentration in neural networks, considered to be the state of the art in supervised learning, as discussed in [18, 29, 30].

The designed systems have been applied to a variety of markets, from Romania [29] to Germany [18] and Brazil [17]. This article focuses exclusively on assets traded in Brazilian stock market Bovespa, although there are no limitations to expand to other markets and other types of assets rather than company stocks.

6. Experiments

The experiments were executed in subset of Brazilian stock market Bovespa. Eight assets with highest trading volume were selected.

Table 1 shows the list of variables collected for the experiment, related to Brazil and its local stock market Bovespa, using the online platforms quandl.com and yahoo.com. The set of features were selected after various iterations of feature selection analysis. The period is from January 2002 to December 2015.

6.1. Trading strategies

Trying out several strategies is computationally expensive, hence we are interested in knowing which are easier to predict than others, in other terms, which ones can be trained

Table 1. Experiment: features

Trading Data
Price Open
Price Close
Price High
Price Low
Volume

Fundamental Anaysis
Selic
Exchange Rate BRL-USD
BM&F Gold Gramme
International Reserves
Foreign Exchange Operations Balance
Bovespa Total Volume
Bovespa Index
Nasdaq Index
Dow Jones Index Index

Technical Analysis
Moving Averages: 10, 20, ... 60 days
Bollinger Bands: 10, 20, ... 60 days

to a higher precision. For the purposes of this experiment, a single strategy is selected after optimizing its parameters.

We define the basic strategy as swing trading, buying and selling an asset when its price increases $X\%$ (success) or the number of days N expires (failure), and optimize for X and N .

In Figure 3, the results of a grid search by two parameters are presented. Profit per day is the expected price increase X divided by number of days N , how much the investment valuates per day in an average case scenario. We will be looking for the highest profit per day which still yields a model with high precision. The trade-off is choosing an optimal point where we have profitable strategies but which are still predictable.

Based on the experiments, we set X to 10 and N to 9, setting the strategy to “sell asset in the next 9 days when its price reaches 10%”.

6.2. Training and Evaluation

The distributed agent platform allows for several agents with similar strategies and assets, but different classifiers, to coexist, since the fund manager will select only the most reliable predictions for trading decisions.

Three different classifiers were tested, Random Forest, Gradient Boosting, and k-Nearest-Neighbors applied on principal components. The results of the experiments are shown in Table 2. The assets are ordered ascending by their trading volume in Bovespa.

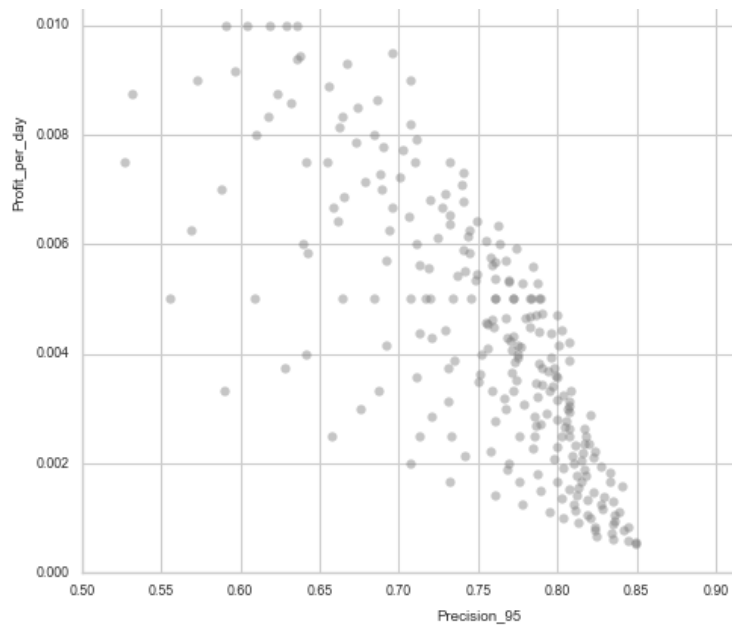


Figure 3. Comparison of profit per day (expected profit / number of days) and precision achieved in the classifier.

Table 2. Experimental Results

Symbol	Positive%	GB-Mean	GB-ME	RF-Mean	RF-ME	kNN-Mean	kNN-ME
ITUB4	0.128	0.865	0.131	0.806	0.119	0.797	0.142
BBDC4	0.142	0.918	0.105	0.906	0.199	0.763	0.116
ABEV3	0.068	0.853	0.153	0.860	0.191	0.787	0.096
PETR4	0.167	0.876	0.107	0.879	0.125	0.791	0.067
VALE5	0.168	0.881	0.132	0.874	0.122	0.806	0.096
VALE3	0.169	0.861	0.117	0.856	0.075	0.806	0.098
BBAS3	0.189	0.889	0.087	0.862	0.105	0.787	0.061
PETR3	0.183	0.858	0.084	0.887	0.114	0.801	0.085

Precision was calculated by using a stratified cross-validation method with 10 folds. The results shown are the average of each validation and the margin of error (ME), given by 2σ . The % of positive labels in each of the training sets is also presented.

Even accounting for the margin of error, the classifier's precision for all assets is above 0.7, for datasets in which the % of positive labels range from 0.06 to 0.19. Gradient Boosting shows the best performance amongst the classifiers.

7. Conclusions

This paper discussed an innovative approach to solve the optimal asset allocation problem in finance investment, aiming to optimize return of investment in a portfolio. It introduces the use of binary classifiers, taking advantage of expert knowledge to build viable strategies, in contrast to traditional stock price prediction approaches using nonlinear regression. Results shown are promising and indicate a new direction of research in the problem

of allocating assets in an investment portfolio.

The distributed computing paradigm in multiagent systems mitigate the increasing cost of training several learning models. The use autonomous agents that can operate independently, in different containers or locations, is also adequate to scalability. Communication protocol standardized in multiagent systems allows the introduction of a diverse set of agents to the system, either electronic agents implemented in different platforms or programming languages or human agents. The fund manager agent implementation is left as a suggestion for follow-up research.

References

- [1] D. Sornette, *Why stock markets crash: critical events in complex financial systems*. Princeton University Press, 2009.
- [2] G. S. Atsalakis, E. M. Dimitrakakis, and C. D. Zopounidis, “Elliott wave theory and neuro-fuzzy systems, in stock market prediction: The wasp system,” *Expert Systems with Applications*, vol. 38, no. 8, pp. 9196–9206, 2011.
- [3] E. F. Fama, “Efficient capital markets: A review of theory and empirical work,” *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.
- [4] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [5] L. O. Souza, C. G. Ralha, and B. W. Hoelz, “Optimizing resource allocation with intelligent agents,” in *Proc. 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. IFAAM, 2017, pp. 1742–1744.
- [6] W. F. Sharpe, “Capital asset prices: A theory of market equilibrium under conditions of risk,” *The Journal of Finance*, vol. 19, no. 3, pp. 425–442, 1964.
- [7] B. Lev and S. R. Thiagarajan, “Fundamental information analysis,” *Journal of Accounting research*, pp. 190–215, 1993.
- [8] H. V. Roberts, “Stock-market patterns and financial analysis: Methodological suggestions,” *The Journal of Finance*, vol. 14, no. 1, pp. 1–10, 1959.
- [9] P. Langley *et al.*, “Selection of relevant features in machine learning,” in *Proc. AAAI Fall symposium on relevance*, vol. 184, 1994, pp. 245–271.
- [10] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.
- [11] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [12] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of statistics*, pp. 1189–1232, 2001.
- [13] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [14] A. Fipa, “Fipa acl message structure specification,” *Foundation for Intelligent Physical Agents*, <http://www.fipa.org/specs/fipa00061/SC00061G.html> (30.6. 2004), 2002.
- [15] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” *Journal of computational science*, vol. 2, no. 1, pp. 1–8, 2011.

- [16] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [17] E. Jabbur, E. Silva, D. Castilho, A. Pereira, and H. Brandão, “Design and evaluation of automatic agents for stock market intraday trading,” *IEEE Computer Society*, pp. 396–403, 2014.
- [18] R. Hafezi, J. Shahrabi, and E. Hadavandi, “A bat-neural network multi-agent system (bnmmas) for stock price prediction: Case study of dax stock price,” *Applied Soft Computing*, vol. 29, pp. 196–210, 2015.
- [19] J. W. Lee, J. Park, O. Jangmin, J. Lee, and E. Hong, “A multiagent approach to q-learning for daily stock trading,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 6, pp. 864–877, 2007.
- [20] Y. Luo, K. Liu, and D. N. Davis, “A multi-agent decision support system for stock trading,” *IEEE network*, vol. 16, no. 1, pp. 20–27, 2002.
- [21] P. A. L. de Castro and J. S. Sichman, “Automated asset management based on partially cooperative agents for a world of risks,” *Applied Intelligence*, vol. 38, no. 2, pp. 210–225, 2013.
- [22] S. Rahimi, R. Tatikunta, R. Ahmad, and B. Gupta, “A multi-agent framework for stock trading,” *International Journal of Intelligent Information and Database Systems*, vol. 3, no. 2, pp. 203–227, 2009.
- [23] X. Gao and L. Chan, “An algorithm for trading and portfolio management using q-learning and sharpe ratio maximization,” in *Proc. International conference on neural information processing*. Citeseer, 2000, pp. 832–837.
- [24] M. A. Dempster and V. Leemans, “An automated fx trading system using adaptive reinforcement learning,” *Expert Systems with Applications*, vol. 30, no. 3, pp. 543–552, 2006.
- [25] J. Moody and M. Saffell, “Learning to trade via direct reinforcement,” *IEEE transactions on neural Networks*, vol. 12, no. 4, pp. 875–889, 2001.
- [26] M. Kearns and L. Ortiz, “The penn-lehman automated trading project,” *IEEE Intelligent Systems*, vol. 18, no. 6, pp. 22–31, Nov 2003.
- [27] A. A. Sherstov and P. Stone, “Three automated stock-trading agents: A comparative study,” in *International Workshop on Agent-Mediated Electronic Commerce*. Springer, 2004, pp. 173–187.
- [28] D. Bloembergen, D. Hennes, S. Parsons, and K. Tuyls, “Survival of the chartist: An evolutionary agent-based analysis of stock market trading,” in *Proc. 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Richland, SC: IFAAM, 2015, pp. 1699–1700.
- [29] M. D. Nemes and A. Butoi, “Data mining on romanian stock market using neural networks for price prediction,” *Informatica Economica*, vol. 17, no. 3, p. 125, 2013.
- [30] T. Chenoweth, Z. Obradovic, and S. S. Lee, “Embedding technical analysis into neural network based trading systems,” *Applied Artificial Intelligence*, vol. 10, no. 6, pp. 523–542, 1996.

Classificação de Gêneros de Música Brasileira usando Letras e Acordes

Dimas Tadeu de Oliveira Junior¹, Tiago Fernandes Tavares¹

¹Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP, Brasil

dimas.oliveira@mail.ru, tavares@dca.fee.unicamp.br

Abstract. *The curation of music lyrics repository works with a huge amount of documents, which can be classified in musical genres, easing its access and indexation. This scenario indicates that an automatic classification system of music lyrics by genre could be useful in the repository organization. The automatic classification system proposed in this paper works by sorting words by its relevancy in relation to a training corpus. The relevancy, measured by a TF-IDF index, compose a feature vector that is delivered to a classification algorithm. The corpus used is formed by 1000 documents, uniformly organized in 10 brazilian music genres. The classification process presented a correct rate of 50%, overcoming results found when using Markov Chains.*

Resumo. *A curadoria de repositórios de letras de músicas lida com grandes quantidades de documentos, que podem ser classificados através de gêneros musicais, facilitando o seu acesso e indexação. Nesse cenário, um sistema automático de classificação de músicas por gênero poderia ser útil na organização dos repositórios. O sistema proposto neste trabalho utiliza o índice TF-IDF para compor um vetor de características que alimenta um algoritmo de classificação. O corpus utilizado contém 1000 documentos, organizados uniformemente em 10 gêneros de música brasileira. O processo de classificação apresentou uma taxa de acerto de 50%, superando os resultados encontrados com o uso de Cadeias de Markov.*

1. Introdução

Com a popularização da internet e dos meios de comunicação em massa o acesso e produção de conteúdo online vem aumentando [1]. Diariamente novos documentos são produzidos e catalogados em sistemas online [2]. O agrupamento das músicas por gêneros musicais facilita o acesso aos documentos, no entanto, quando este tipo de trabalho é realizado manualmente, a tarefa se torna pouco convidativa, pois além do grande volume de documentos¹ existe também uma infinidade de gêneros musicais a serem considerados [3].

O agrupamento de músicas em gêneros musicais é uma tarefa que leva em consideração elementos subjetivos [4, 5]. Por esse motivo, gêneros diferentes podem ser atribuídos à mesma música, caso ela seja classificada por pessoas diferentes [6]. Além disso, a própria definição dos gêneros musicais pode ser ambígua, de forma que não é possível determinar com clareza o gênero a qual pertence uma determinada música [7].

¹Existem, por exemplo, mais de quatrocentas mil músicas disponíveis no portal Cifra Club

A classificação automática de músicas é uma tarefa bastante conhecida no campo da Recuperação de Informação Musical (*Music Information Retrieval* - MIR). O gênero musical é um importante descritor do conteúdo musical [4] utilizado pela indústria musical e seus consumidores na organização de coleções musicais [3] [6].

Duas abordagens para a classificação de gêneros musicais são a análise de sinais sonoros e a análise de símbolos [8][9]. Grande parte das pesquisas que tratam desse assunto não fazem uso das letras. No entanto, cada vez mais têm surgido trabalhos relacionando informações extraídas a partir das letras com as informações encontradas nos sinais de áudio [10][11].

O uso das letras musicais em conjunto com os sinais sonoros tende a melhorar resultados de classificação quando comparados com classificadores que usam somente análise de áudio [2]. Diversas características podem ser extraídas das letras tais como o tema, sentimentos, rimas e o tamanho dos versos [8].

Outro importante aspecto de músicas populares é a sua transcrição na forma de cifras para violão. As cifras são representações que evidenciam a relação entre os acordes que devem ser tocados para acompanhar a letra. O termo cifra também se aplica aos símbolos utilizados na representação de um acorde, onde letras do alfabeto determinam a nota fundamental (Ex.: C = Dó, D = Ré, E = Mi, etc) e números indicam os intervalos utilizados na construção do acorde a partir da nota fundamental (Ex. C7: o número 7 faz referência ao intervalo de sétima menor a partir da fundamental Dó) [12]. Os acordes são comumente encontrados através da audição por pessoas com conhecimento em música.

É possível representar as progressões dos acordes usando modelos estocásticos e, à partir desses modelos, realizar previsões sobre os gêneros musicais correspondentes [9]. Gramáticas geradas através de lógicas de primeira ordem também podem ser utilizadas para determinar, de forma automática, o gênero musical à partir dos acordes empregados em uma determinada música [1].

Modelos Ocultos de Markov (*Hidden Markov Model* - HMM) [13][14] já foram utilizados para a geração de modelos específicos por gêneros musicais na tarefa de transcrição de acordes a partir de arquivos de áudio [15]. Nesta aplicação, o HMM busca capturar as probabilidades de transições de acorde típicas de cada gênero. O uso de HMMs não se limita a acordes, podendo também ser utilizado na classificação de conteúdo textual [16].

Uma outra forma comum para a classificação de documentos textuais é usar um índice criado usando as palavras encontradas em um conjunto de documentos. Este índice é formado pela frequência dos termos (*Term Frequency* - TF) e pelo inverso da frequência dos termos através de todos os documentos (*Inverse Document Frequency* - IDF). O produto destes dois cálculos (TF-IDF) mede a relevância das palavras em um documento, em relação a uma coleção de documentos [17]. O critério TF-IDF também pode ser usado para a construção de modelos *bag-of-words* (BoW). Nesses modelos, os documentos são representados por vetores onde cada posição de um vetor representa o índice TF-IDF de uma palavra diferente [18].

Este trabalho apresenta um estudo comparativo entre esses diferentes métodos – TF-IDF e modelo estatístico de primeira ordem – e de diferentes entradas – acordes e informação textual – na tarefa de classificação automática de músicas brasileiras por

gênero. Na seção 2, apresentamos os métodos de classificação. A construção da base de dados é apresentada na seção 3 e os resultados obtidos são descritos na seção 4. Por fim, na seção 5 tratamos da conclusão e traçamos direções para futuras pesquisas.

2. Método

Neste trabalho, utilizamos dois métodos distintos para classificação. O primeiro método emprega HMMs para gerar modelos estatísticos tanto para as letras – representadas como sequências de caracteres ou como sequências de palavras, em testes distintos – como para os acordes. O segundo método estima a relevância de palavras usando o índice TF-IDF. O índice é usado para criar um vetor de palavras (*bag-of-words*).

Um modelo semelhante ao *bag-of-words* foi usado para estimar a relevância de acordes. Nesse modelo, chamado *bag-of-chords* (BoC), sequências de acordes são representados usando o grau do acorde em relação a tonalidade da música.

As subseções a seguir descrevem as técnicas e conceitos utilizados.

2.1. Cadeias de Markov

Neste trabalho, utilizamos três diferentes modelos baseados em Cadeias de Markov: modelos de caracteres, de palavras e de acordes. Os modelos de caracteres utilizam 27 símbolos: letras de A a Z mais um símbolo para o espaço. Esse modelo é representado por uma matriz 27×27 , onde as transições entre os caracteres encontrados nas letras são contabilizados e normalizados. Os modelos de palavras são representados por matrizes esparsas e os modelos de acordes (graus) são representados por matrizes 24×24 (12 graus maiores e 12 graus menores). Os parâmetros relacionados a cada gênero musical foram representado por 3 modelos distintos, um para cada tipo de atributo, como mostra a Figura 1.

2.2. Bag-of-Words e TF-IDF

No modelo *bag-of-words*, um conjunto de palavras forma um vetor que é capaz de representar diferentes documentos num espaço vetorial padrão. Neste tipo de modelo, a ordenação das palavras, bem como suas relações gramaticais, são descartadas. Cada palavra no vetor é considerada como uma característica do documento.

É possível utilizar todas as palavras do *corpus* na criação do vetores *bag-of-words*. No entanto, isso resultaria em um vetor de características com muitas dimensões. Uma forma de reduzir as dimensões do vetor é através da exclusão de palavras pouco significativas (palavras muito raras ou muito comuns). A seleção das palavras mais relevantes se dá através do índice TF-IDF, onde são considerados as frequências de ocorrências dos termos e também as quantidades de documentos onde estes termos aparecem. O índice TF-IDF foi calculado conforme a expressão:

$$\text{TF-IDF} = TF \times \log \frac{nD}{nDF} \quad (1)$$

onde TF é a frequência do termo no documento, nD representa o número de documentos utilizados e nDF contabiliza a quantidade de documentos em que o termo ocorreu.

2.3. Acordes e Graus

Um acorde é formado por pelo menos três notas que podem ser ouvidos simultaneamente. A sequência de acordes presentes em uma música é denominada progressão de acordes. Alguns gêneros musicais apresentam progressões típicas de acordes (tônica - subdominante - dominante), o que pode ser utilizado como característica para diferenciar as músicas em seus respectivos gêneros musicais.

Uma forma de representação de acorde muito utilizada na música popular é chamada cifra. Uma cifra é composta por uma letra (A=Lá, B=Si, C=Dó, D=Ré, E=Mí, F=Fá, G=Sol), podendo incluir também símbolos e números (Ex.: Gm6, D9b, A#7), os quais representam os intervalos entre os sons que compõem o acorde a partir da nota fundamental utilizada (tônica ou dominante, primeiro grau).

Na análise harmônica é comum o uso de algarismos romanos para representar os graus dos acordes. Nessa representação, as letras que identificam os acordes são substituídas pelos algarismos (Ex.: C-F-G → I-IV-V), levando em consideração a tonalidade presente na música. Essa forma de representação permite que progressões de acordes de músicas em diferentes tons possam ser comparados de forma mais fácil, além de diminuir o número de dimensões necessárias para representar as possíveis progressões de acordes nas músicas.

Neste trabalho os acordes foram simplificados de forma a envolver somente as tríades maiores e menores [19]. As tríades maiores e menores são formadas pela nota fundamental, a terça (maior ou menor, respectivamente) e a quinta justa. A partir dessa simplificação e do uso dos graus para se referir aos acordes, passamos a ter 24 acordes possíveis (12 maiores e 12 menores) para representar as músicas analisadas.

2.4. Extração de características

No primeiro método a extração das características é realizada em duas fases: pré-treino e treino. No pré-treino as Cadeias de Markov são geradas utilizando parte das amostras do conjunto de treino. Na fase de treino, os vetores de características são calculados a partir de Cadeias de Markov utilizando a outra parte das amostras do conjunto de treino. Para cada atributo (caracteres, palavras ou graus) são calculadas as probabilidades da amostra pertencer a cada um dos gêneros musicais analisados. Os valores das probabilidades obtidas para cada atributo/gênero são armazenadas em um vetor conforme a Figura 2. Os vetores gerados, juntamente com as referências aos gêneros das amostras são utilizados para treinar um classificador SVM. O modelo de treino obtido pelo classificador é usado posteriormente para prever o gênero musical das amostras de teste.

No segundo método, o modelo *bag-of-words* gerado a partir do índice TF-IDF é utilizado para representar as amostras. A ocorrência dos termos foi anotada de forma binária, ou seja, apenas a presença ou ausência dos termos encontrados são considerados. Esse método conta também com o modelo *bag-of-chords*, representando de forma binária a ausência ou presença de determinado grau na sequência de acordes presentes na música. Os vetores gerados pelos dois atributos são combinados em um vetor apenas (ver Figura 3) para então serem entregues ao classificador. Os experimentos são descritos em detalhe na Seção 3.

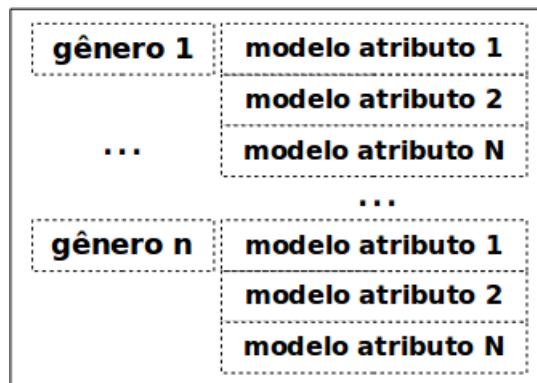


Figura 1. Modelos baseados em atributos, agrupados por gênero musical

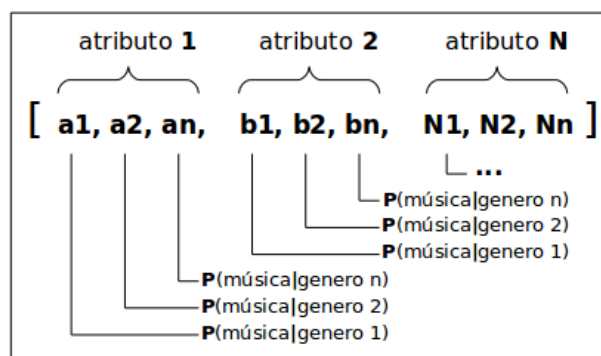


Figura 2. Vetor de características formado pelas respostas dos modelos markovianos

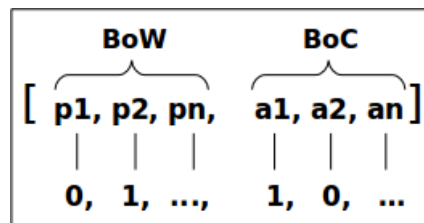


Figura 3. Vetor de características com *bag-of-words* (BoW) e *bag-of-chords* (BoC)

2.5. Classificação

Diferentes classificadores foram utilizados no problema de classificação de gêneros musicais: *K-Nearest Neighbors* (KNN), *Random Forest* (RF), *Logistic Regression* (LR) e *Support Vector Machine* (SVM).

Os parâmetros do SVM foram encontrados através de um procedimento de busca exaustiva, visando maximizar o índice de acertos numa fração (20%) do conjunto de teste. Os valores dos parâmetros obtidos estão listados na Tabela 1.

3. Experimentos

Esta seção descreve em detalhes os experimentos realizados, bem como a base de dados coletada.

Tabela 1. Parâmetros utilizados no classificador SVM

Método	Kernel	C	Gamma
HMM (1)	linear	1	0.01
TF-IDF (2)	rbf	100	0.01

3.1. Base

Foram coletadas 1000 músicas brasileiras de um portal de cifras de músicas brasileiras² através de técnicas de *web scrapping*. As músicas foram organizadas uniformemente a partir de um subconjunto dos gêneros já utilizados pelo portal. Os gêneros escolhidos para este trabalho foram 10: axé, bossa-nova, brega, funk carioca, gospel religioso, MPB, reggae, rock, samba e sertanejo. As letras tiveram os sinais diacríticos removidos e as palavras foram capitalizadas. Os acordes foram extraídos a partir do código HTML e as letras foram separadas dos acordes. Ao todo, a base construída envolveu 428 artistas, num total de 161.693 palavras (9823 palavras únicas).

3.2. Métrica e Protocolo

Os resultados obtidos nos dois métodos foram avaliados através da medida *F1-score* (F1), que é a média harmônica entre as medidas *precision* (P) e *recall* (R), que são definidas pelas seguintes equações:

$$P = \frac{c}{c + f^+} \quad (2)$$

$$R = \frac{c}{c + f^-} \quad (3)$$

$$F1 = 2 \times (P \times R) \div (P + R) \quad (4)$$

onde c corresponde ao número de documentos recuperados corretamente, f^+ indica a quantidade de falsos positivos e f^- corresponde a quantidade de falsos negativos.

Os experimentos seguiram um protocolo de validação cruzada em 10 dobras (*10-fold cross validation*).

No primeiro método (HMM) as amostras foram divididas em três partes: 45% para geração dos modelos markovianos, 45% para treino do classificador e 10% das amostras para testes. Experimentos foram realizados para verificar a contribuição dos atributo de forma isolada e também em conjunto.

O segundo método (TF-IDF) teve a base dividida em duas partes: 90% das amostras para a geração dos modelos *bag-of-words* e *bag-of-chords* (treino) e os 10% restantes da base foram utilizados na realização dos testes. Experimentos com variações entre 100 e 8000 palavras selecionadas pelo índice *TF-IDF* foram realizados.

²www.cifraclub.com.br

4. Resultados e Discussão

As tabelas 2, 3 e 4 mostram os resultados de classificação médios para cada gênero, obtidos através do classificador SVM. A Tabela 5 mostra os resultados obtidos com outros classificadores, com valores inferiores aos encontrados pelo classificador SVM. Como é possível verificar, o método que utiliza TF-IDF apresentou resultados superiores ao método que utiliza Cadeias de Markov. Não houve uma variação significativa entre 4000 e 8000 palavras, os resultados aqui registrados foram obtidos com modelos *bag-of-words* de 4000 palavras. O uso combinado dos atributos em ambos os métodos trouxe uma pequena melhora nos resultados gerais. Os menores índices de acerto ocorreram nos modelos markovianos ao se utilizar os atributos separadamente.

A matriz de confusão, mostrada na Figura 5, mostra que os erros estão concentrados na confusão entre certos gêneros, como por exemplo os gêneros bossa-nova e samba. Também, é possível verificar que muitas amostras são classificadas erroneamente como MPB.

Em nenhum dos métodos, o uso dos acordes sozinhos foi suficiente para predizer os gêneros musicais das amostras. No entanto, alguns gêneros musicais se beneficiaram da utilização dos acordes como características adicionais ao uso das letras.

Tabela 2. HMM - F1-score para os atributos utilizados separadamente

	Graus	Caracteres	Palavras
Axé	0.14	0.04	0.24
Bossa-nova	0.40	0.24	0.30
Brega	0.43	0.06	0.25
Funk Carioca	0.29	0.06	0.36
Gospel	0.37	0.24	0.81
MPB	0.18	0.04	0.04
Reggae	0.13	0.02	0.20
Rock	0.22	0.00	0.29
Samba	0.43	0.12	0.31
Sertanejo	0.26	0.27	0.25
Média (μ)	0.28	0.11	0.31
Desvio Padrão (σ)	0.04	0,02	0,04

O gênero gospel foi o que apresentou o melhor F1-score médio, o que provavelmente está ligado aos temas religiosos tratados nas letras, retratado em palavras específicas relacionadas a esse tipo de tema. Também é possível verificar na Tabela 4 que a harmonia das músicas gospel não melhoraram sua acurácia, o que reforça a ideia de que o tema recorrente nas letras das músicas gospel foram mais marcantes que a harmonia empregada. Isso indica também um uso de progressões de acordes mais simples neste gênero.

O funk carioca foi o segundo melhor classificado (método baseado em TF-IDF), assim como no gospel, o uso dos acordes parece não trazer melhorias na sua classificação. O tema encontrado no funk carioca geralmente trata das comunidades carentes do Rio de Janeiro e seus problemas.

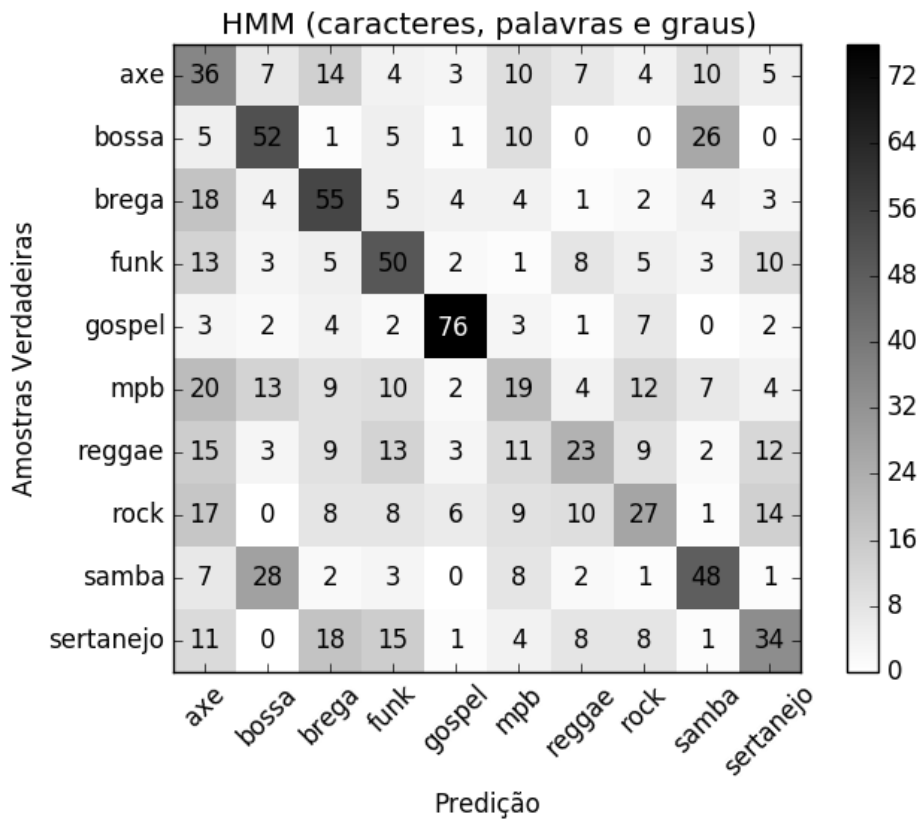


Figura 4. Matriz de confusão utilizando modelos markovianos

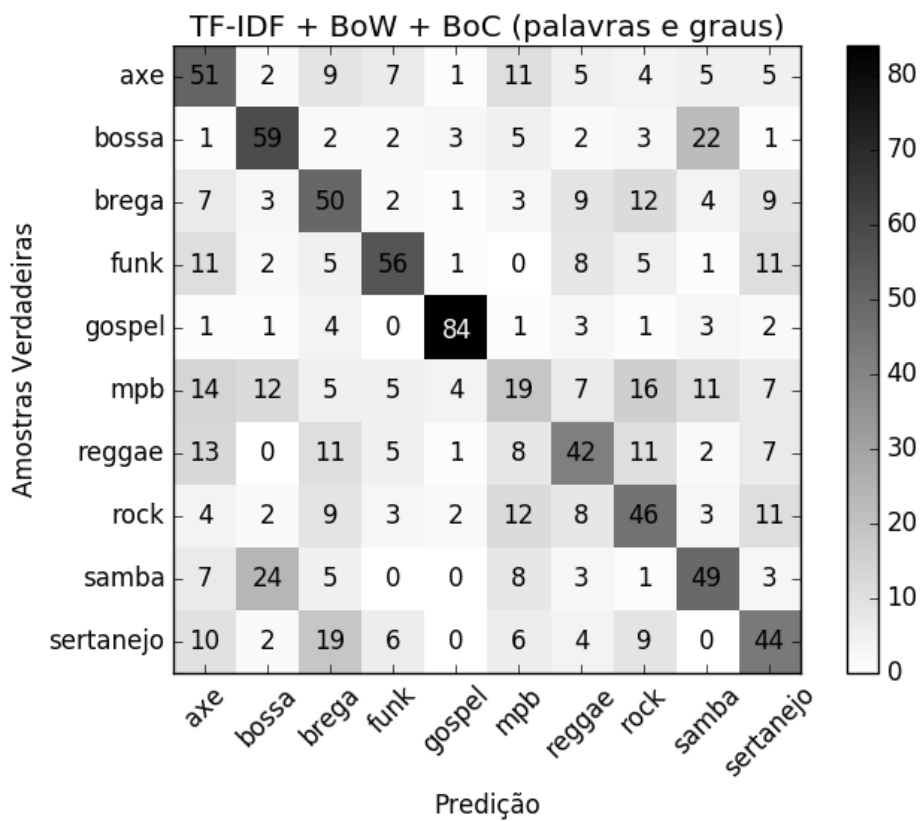


Figura 5. Matriz de confusão utilizando TF-IDF, BoW e BoC

Tabela 3. HMM - F1-score para atributos combinados

	Graus Caracteres	Graus Palavras	Graus Caracteres Palavras
Axé	0.27	0.29	0.29
Bossa-nova	0.42	0.47	0.49
Brega	0.42	0.45	0.49
Funk Carioca	0.35	0.42	0.47
Gospel	0.42	0.77	0.77
MPB	0.15	0.23	0.21
Reggae	0.16	0.31	0.28
Rock	0.20	0.30	0.31
Samba	0.44	0.47	0.48
Sertanejo	0.27	0.38	0.37
Média (μ)	0.31	0.41	0.42
Desvio Padrão (σ)	0.05	0,05	0,03

Tabela 4. TF-IDF - F1-score para atributos separados e combinados

	Graus	Palavras	Graus Palavras
Axé	0.16	0.45	0.47
Bossa-nova	0.45	0.45	0.57
Brega	0.37	0.32	0.46
Funk Carioca	0.31	0.60	0.60
Gospel	0.25	0.84	0.85
MPB	0.11	0.16	0.22
Reggae	0.15	0.35	0.44
Rock	0.21	0.38	0.44
Samba	0.40	0.47	0.49
Sertanejo	0.30	0.41	0.44
Média (μ)	0.27	0.45	0.50
Desvio Padrão (σ)	0.02	0.02	0,04

O gênero MPB gerou bastante confusão, sendo confundido de forma parecida com 3 outros gêneros (rock, axé e bossa nova). Essa confusão condiz com a evolução do gênero MPB, que, no decorrer da sua história, passou a englobar outros gêneros musicais.

As Cadeias de Markov não foram suficientes para classificar adequadamente os gêneros axé, MPB, reggae, rock e sertanejo (ver Tabela 3), no entanto, para os outros gêneros, os resultados obtidos se parecem proporcionalmente com os valores encontrados no segundo método de classificação (TF-IDF).

O uso de características harmônicas foi mais marcante em alguns gêneros, como a bossa nova por exemplo, onde os acordes junto as características textuais aumentou em 10% seu resultado (Tabela 4), o que pode ser explicado pelo uso de harmonias mais

Tabela 5. F1-score obtido utilizando K Nearest Neighbors, Random Forest e Logistic Regression.

	HMM			TF-IDF		
	Graus Caracteres Palavras			Graus e Palavras		
	KNN	RF	LR	KNN	RF	LR
Axé	0.24	0.25	0.31	0.22	0.24	0.44
Bossa-nova	0.37	0.38	0.48	0.30	0.44	0.57
Brega	0.26	0.30	0.46	0.18	0.32	0.44
Funk Carioca	0.31	0.29	0.48	0.09	0.35	0.56
Gospel	0.42	0.46	0.78	0.41	0.67	0.83
MPB	0.07	0.16	0.23	0.03	0.20	0.20
Reggae	0.12	0.16	0.29	0.02	0.30	0.41
Rock	0.21	0.17	0.32	0.14	0.23	0.45
Samba	0.37	0.31	0.43	0.05	0.41	0.50
Sertanejo	0.19	0.30	0.36	0.00	0.25	0.46
Média (μ)	0.26	0.28	0.41	0.15	0.34	0.49
Desvio Padrão (σ)	0.03	0.03	0.02	0.02	0.03	0.02

complexas nesse gênero. A falta de um modelo de acordes mais completo pode ser responsável pela pequena contribuição desse tipo de modelo nos resultados.

5. Conclusão

Este trabalho analisa os resultados da aplicação de diferentes métodos de classificação de música por gênero em uma base de dados de letras e acordes de músicas brasileiras. Os resultados mostram que a relevância de letras e de acordes são diferentes para a identificação de diferentes gêneros musicais.

Os resultados obtidos demonstram que a informação fornecida por letras e acordes é relevante para a classificação, embora ainda haja espaço significativo para melhorias. O modelo de letras pode ser expandido para levar em consideração a estrutura do texto em versos, padrões de rimas, tamanho de frases e a complexidade dos textos. O modelo harmônico, por sua vez, pode passar a empregar tétrades ao invés de tríades, uma vez que existem estudos mostrando que o uso de modelos harmônicos mais completos podem trazer melhores resultados. Esses aspectos serão explorados em trabalhos futuros.

Referências

- [1] Amélie Anglade, Rafael Ramirez, Simon Dixon, et al. Genre classification using harmony rules induced from automatic chord transcriptions. In *ISMIR*, pages 669–674, 2009.
- [2] Michael Fell and Caroline Sporleder. Lyrics-based analysis and classification of music. In *COLING*, volume 2014, pages 620–631, 2014.
- [3] François Pachet and Daniel Cazaly. A taxonomy of musical genres. In *Content-Based Multimedia Information Access-Volume 2*, pages 1238–1245. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2000.

- [4] Pouya Ghaemmaghami and Nicu Sebe. Brain and music: Music genre classification using brain signals. In *Signal Processing Conference (EUSIPCO), 2016 24th European*, pages 708–712. IEEE, 2016.
- [5] J Zheng and M Oussalah. Automatic system for music genre classification, university of birmingham. *Electronics, Electrical and Computer Engineering*, 2006.
- [6] Cory McKay and Ichiro Fujinaga. Automatic genre classification using large high-level musical feature sets. In *ISMIR*, volume 2004, pages 525–530. Citeseer, 2004.
- [7] Tiago Filipe Beato Mourato de Matos. Métodos estatísticos de classificação de géneros musicais. 2013.
- [8] Rudolf Mayer, Robert Neumayer, and Andreas Rauber. Rhyme and style features for musical genre classification by song lyrics. In *ISMIR*, pages 337–342, 2008.
- [9] Carlos Pérez-Sancho, David Rizo, and José M Inesta. Genre classification using chords and stochastic language models. *Connection science*, 21(2-3):145–159, 2009.
- [10] Dawen Liang, Haijie Gu, and Brendan O’Connor. Music genre classification with the million song dataset. *Machine Learning Department, CMU*, 2011.
- [11] Xiao Hu and J Stephen Downie. Improving mood classification in music digital libraries by combining lyrics and audio. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 159–168. ACM, 2010.
- [12] Almir CHEDIAK. Harmonia e improvisação: 70 músicas harmonizadas e analisadas. *Violão, baixo e teclado. Rio de*, 1986.
- [13] Kwan Yi and Jamshid Beheshti. A text categorization model based on hidden markov models. In *Proceedings of the Annual Conference of CAIS/Actes du congrès annuel de l’ACSI*, 2013.
- [14] Brian Hayes et al. First links in the markov chain. *American Scientist*, 101(2):252, 2013.
- [15] Kyogu Lee. A system for automatic chord transcription from audio using genre-specific hidden markov models. In *International Workshop on Adaptive Multimedia Retrieval*, pages 134–146. Springer, 2007.
- [16] Kwan Yi and Jamshid Beheshti. A hidden markov model-based text classification of medical documents. *Journal of Information Science*, 35(1):67–81, 2009.
- [17] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.
- [18] Adriano A de Lima, Rodrigo M Nunes, Rafael P Ribeiro, and Carlos N Silla Jr. Nordic music genre classification using song lyrics. In *International Conference on Applications of Natural Language to Data Bases/Information Systems*, pages 89–100. Springer, 2014.
- [19] Matthias Mauch. Automatic chord transcription from audio using computational models of musical context. 2010.

Multi-Task Reinforcement Learning: An Hybrid A3C Domain Approach

Marco A. F. Birck, Ulisses B. Correa, Pedro Ballester,
Virginia O. Andersson, Ricardo M. Araujo

¹PPGC - Federal University of Pelotas (UFPEL),
Pelotas, RS, Brazil

{mafbirck, ub.correa, plballester, vandersson, ricardo}@inf.ufpel.edu.br

Abstract. *In this paper we evaluate the capabilities of the Asynchronous Advantage Actor-Critic (A3C) reinforcement learning algorithm for multi-task learning, where a single model is asked to solve multiple tasks at once. Learning multiple tasks is hard due to the increased complexity, typically requiring deeper networks and modular systems. We show that when there is semantic similarity between tasks, the A3C algorithm is able not only to learn both tasks simultaneously, but is also able to improve the model's performance due to knowledge transfer between tasks.*

1. Introduction

Multi-task Reinforcement Learning [Taylor and Stone 2011] is a growing field that aims at learning multiple domain tasks with a single model. As with many other machine learning algorithms, Multi-task Reinforcement Learning is inspired by cognitive capabilities of the brain, in particular the ability to adapt and leverage knowledge efficiently in multiple complex contexts, an important feature in many areas such as medicine, robotics, and data analysis.

Deep Reinforcement Learning (DRL) has become an important research area, where deep neural networks are applied to reinforcement learning problems, reaching state-of-the-art results in several tasks [Mnih et al. 2015, Lillicrap et al. 2015, Silver et al. 2016, Mnih et al. 2016]. Several proposals have been made to address multi-task learning in DRL. One such proposal is to train a network on a particular task and then fine-tune the resulting model on successive tasks. This naive approach typically results in models that degrade performance on previous task, as the network "forgets" what was learned, whether no mechanism to avoid it is present [Kirkpatrick et al. 2017]. More effective approaches are also more complex, modifying the training algorithm or proposing different networks.

In this work, we provide an initial exploration of the A3C (Asynchronous Advantage Actor-Critic) DRL algorithm applied to multi-task learning. A3C was proposed by [Mnih et al. 2016] and provides a multi-threaded and asynchronous approach to DRL. This algorithm allows a model to be trained using multiple, different explorations of a single target task, providing data sparsity and avoiding the use of memory replay. Given that multi-threading characteristic, we evaluate the A3C ability to perform multi-task learning without modifications when applied to different, but semantically related tasks. To do so,

we simultaneously train multiple tasks using a single A3C model, allowing the network to asynchronously share knowledge obtained from and to all tasks. We report on results from experiments where a single A3C model is asked to learn two different tasks. Our approach is to split the running threads and let each task take one half of the threads - e.g. two threads running Task A and two threads running Task B.

We use two pairs of games from the Arcade Learning Environment (ALE) as testbeds, selecting games that share elements and general mechanics. Our results shows that for these pairs not only the network is able to learn both tasks simultaneously, but convergence speed for some tasks increases substantially. We argue that conducting multi-task learning using A3C can be beneficial even when the interest lies in training on a single task.

The remaining of this paper is organized as follows. Section 2 discusses related work on Deep Reinforcement Learning and Multi-task learning. Section 3 presents our approach and methodology. Section 4 our experimental results are shown and analyzed. Finally, Section 5 summarizes the main contributions and indicates further lines of research.

2. Related works

Over the last few years growing attention has been put over Deep Reinforcement Learning (DRL), as it has become the state-of-the-art in several benchmark tasks and real-world problems. In this section we will discuss recent relevant techniques to multi-task DRL presented in related works.

Distillation and Actor-Mimic. Policy Distillation (PD) [Rusu et al. 2015] and Actor-Mimic (AM) [Parisotto et al. 2015] use the concept of distillation as a possible approach to multi-task DRL. Distillation was first suggested by [Buciluă et al. 2006], aiming at minimizing computational costs of ensemble methods. The authors propose a method in which, given an ensemble of baselines models, a new smaller model is constructed, allowing for application in real scenarios. Then, [Hinton et al. 2015] mapped this concept to the Neural Networks domain. In PD and AM the distillation procedure was used since, given a set of experts and teachers on a task, a DQN [Mnih et al. 2015] could distill the knowledge by reusing the inputs and activations. Each trained task uses the same feature extraction neural network, but with different controllers. Results show this approach have potential for multi-task learning, although the methodology requires an expert on the domain.

Progressive Neural Networks. In [Rusu et al. 2016], a model was proposed using the concept of lateral features transferring. The author's proposed model has the ability to learn new tasks and maintain previous knowledge using Progressive Neural Networks. At each new task, the model learns a new column - which can be seen as a new neural network - that will be frozen and used in successive tasks. Although successfully solving many multi-task learning problems, the model is computationally expensive, as it grows with each new task presented.

PathNet. Another approach to multi-task reinforcement learning is PathNet [Fernando et al. 2017], which is able to learn multiple tasks by training a set of N tasks, one at a time. Their algorithm uses the concept of evolutionary theory, in which paths inside a network are learned and added to handle specific tasks. These paths, called genotypes, can be useful for other tasks and are thus shared between them.

Asynchronous Advantage Actor-Critic (A3C). The A3C algorithm [Mnih et al. 2016] uses training agents to take advantage of parallelism jointly with the actor-critic method, gaining performance. The algorithm uses a single global network that the different and independent threads can update asynchronously. This update is conducted after a determined number of steps (typically $t_{max}=20$ steps), when gradients are accumulated and used to perform the update. By training different task instances simultaneously, the algorithm adds sparsity to the training data and removes the need of a memory replay, a requirement present in pioneer works such as [Mnih et al. 2013] and [Mnih et al. 2015]. Applications of A3C surpass the current state-of-art on the ALE domain, in addition to reducing significantly training times without the need of GPU.

3. Goals and Methodology

Our main goal is to understand how A3C handles multi-task learning and the impact this has on individual tasks. We make use of the parallel learning nature of the algorithm to allow multiple domain environments to be trained jointly. In order to do so, we launch multiple agents (threads) with different tasks and observe the effects of this interplay compared to when only a single task is used. Through this multi-environment procedure, our approach tries to surpass the high computational needs involved in other multi-task approaches whilst giving a much simpler architectural solution. Our hypothesis is that the single shared model of A3C, when updated asynchronously over different tasks, is able to conduct multi-task learning and effectively learn all tasks. We are interested in understanding the effect of such learning in the individual tasks.

We name our approach Hybrid Asynchronous Advantage Actor-Critic Training (Hybrid A3C), given its nature of learning multiple tasks online and jointly. A3C, the basis of our approach, is a multi-thread and asynchronous algorithm using the actor-critic model. The main reason for using multiple threads is to provide the model with different scenes of a single task. To test the multi-task capabilities of A3C, we created a different training pipeline, as shown in Figure 1. Instead of the usual multiple threads with the same task, we train the algorithm by simultaneously providing different tasks to the training threads. The same number of training threads is used for each task, thus avoiding that one task take control of the global network model. Our approach, compared to other multi-task methods discussed previously, has much less architectural requirements, discards the need for master pre-trained agents and at the same time leverages the fact that A3C converges much faster than alternative methods.

To evaluate such approach, we use games from the Atari Learning Environment (ALE) provided by the OpenAI Gym [Brockman et al. 2016] as benchmark scenarios. We pick pairs of games that present similar mechanics and elements. We require them to be similar so as to allow for knowledge to be shared. For example, we assume the games Pong and Breakout represent a similar environment as both of them consist mainly

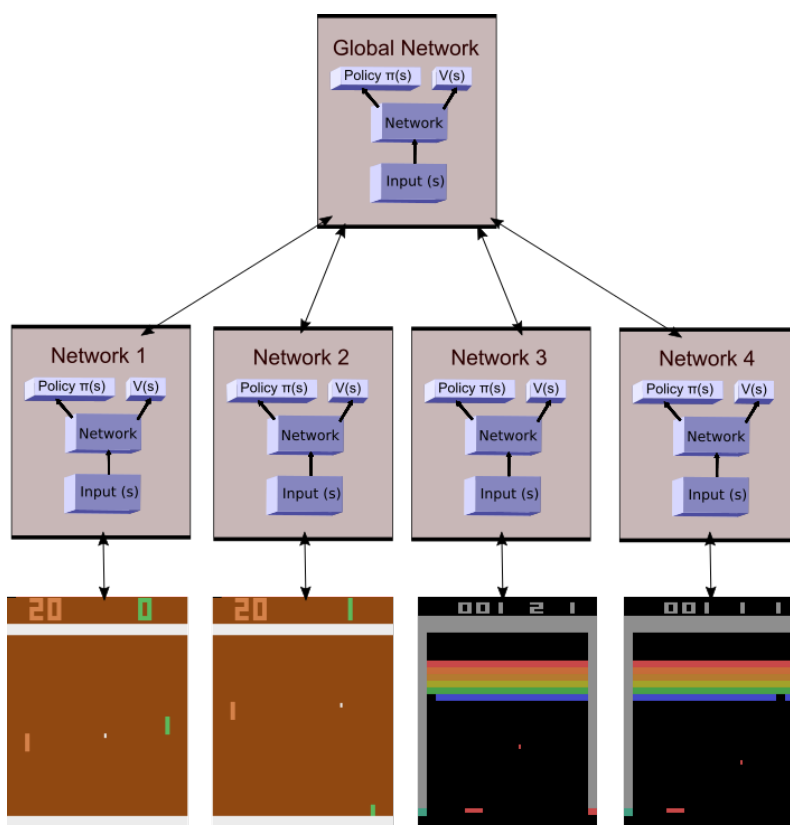


Figure 1. Hybrid A3C training Pipeline.

of a side-to-side moving paddle that must hit a ball properly. Evaluation is performed by observing how reward obtained per unit of time evolves over several episodes for each task using the hybrid model and compare the results against training using a single task.

We use an open-source version of A3C¹, which differs slightly from the original A3C proposal. In the used implementation, two branches neural networks, a policy network $\pi(a_t|s_t; \theta)$ and a estimate value function $V(s_t; \theta_v)$ are used. They share a feature extractor composed of four convolutional layers consisting of 32 filters of 3x3 size, stride 1, connected to a 256 LSTM cells layer. LSTM outputs are sent to the policy and value networks. We adopted Adam optimizer instead of RMSProp optimizer to make implementation simpler and used a fixed learning rate $\alpha = 10^{-4}$. We do not conduct hyper-parameter tuning.

4. Experiments

In this section we conduct a set of experiments aiming at understanding how training A3C on multiple tasks affect the performance on each individual task. From the ALE set, we select two pairs of games that share similar mechanics and elements: (i) Pong and Breakout; (ii) Space Invaders and Demon Attack.

Figure 2 shows screenshots of the selected games. In both Pong and Breakout, a player must control a paddle in order to hit a ball. For Pong the player must attempt to make an opponent miss the ball, while for Breakout the goal is to break as many bricks

¹<https://github.com/openai/universe-starter-agent>

as possible. Similarly, for both Space Invaders and Demon Attack a player must control a ship that can shoot and must hit enemies. For Space Invaders, enemies are more passive, moving in a regular fashion while for Demon Attack there is a greater variety of enemies, most of them shooting back and moving in more erratic patterns. Even though games in each pair share elements between them, they cannot be simply reduced to one another (i.e. seen as the same game) as they have unique scoring mechanisms and features.

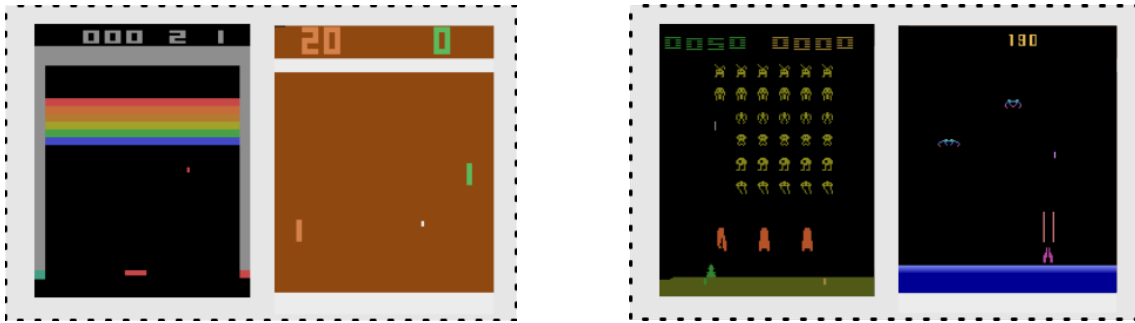


Figure 2. Screenshots for the four selected games. From left to right: Breakout, Pong, Space Invaders, Demon Attack.

For each pair, we conduct two experiments: train the game individually and jointly. In each case the same number of threads (two) is allocated for each individual game and training is conducted for 50 million time-steps.

Figure 3 shows the results for the pair Pong and Breakout. In Figure 3 (a), results for Breakout trained jointly with Pong and independently are shown. It is possible to see that when training jointly the model is able to obtain much higher rewards when compared to the standalone version. A similar behavior is observed in Figure 3 (b), where results for Pong are shown; for Pong joint training yields worse initial rewards but the model converges to a better policy. However, the difference is not nearly as pronounced as the one observed for Breakout and both models behave very similarly.

Figure 4 shows the results for the pair Space Invaders and Demon Attack. For Demon Attack (Figure 4 (b)), we can observe that the hybrid approach leads to a significant improvement in performance. On the other hand, a mixed performance is observed for Space Invaders, with rewards remaining somewhat the same in both hybrid and standalone versions.

These results show that A3C can handle multi-task learning, at least in the case where there are two tasks that are somewhat similar. More striking is how performance is *improved* in some cases; both Breakout and Demon Attack benefit from being trained jointly with Pong and Space Invaders, respectively. When observing each game, both Breakout and Demon Attack can be seen as more complex games compared to Pong and Space Invaders. This leads to the observation that one possible explanation is that the harder task benefits from knowledge obtained from the easier one. The easier tasks, however, are slightly worse off by the presence of the harder tasks.

5. Conclusion and Discussion

In this work we reported on initial experiments applying the DRL algorithm A3C to a multi-task scenario. By splitting the running threads among different tasks, we showed

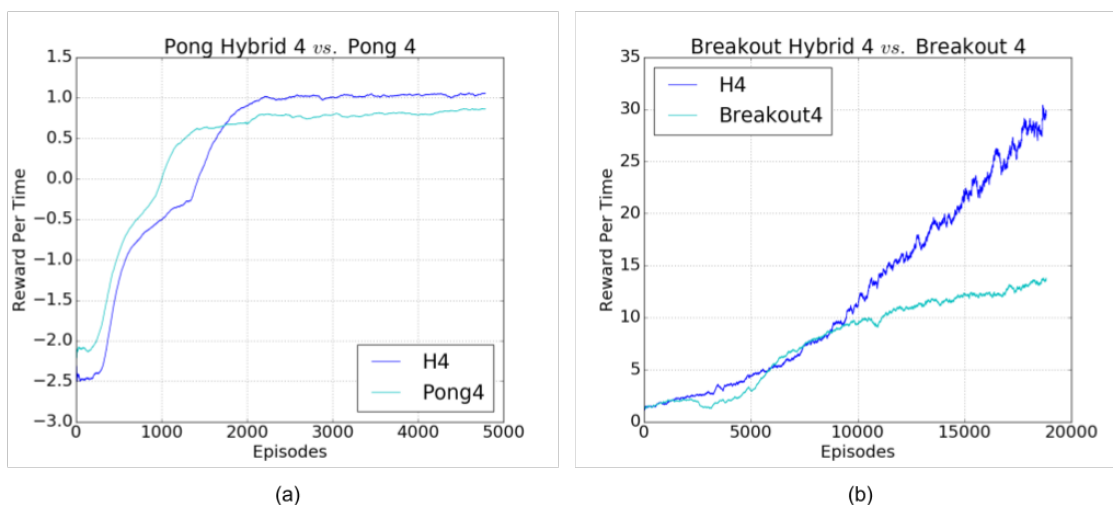


Figure 3. Results for standalone and joint training for (a) Breakout and (b) Pong. H4 stands for the hybrid approach, with two threads being used for each game, while Breakout4 and Pong4 stands for the standalone cases using four threads dedicated to sole game. Curves were smoothed using Exponential Weighted Moving Average in order to better observe general trends.

that the algorithm is able to simultaneously learn two games from ALE without degradation in performance for any one task. On the contrary, we noticed an *improvement* in performance for one task in each tested pair; the improvement appears to have occurred on the more complex task, while for the simpler one mixed results were observed, with little difference compared to training the task alone.

While further experiments are required to draw strong conclusions, one possible explanation for the observed behavior is that of curriculum learning [Bengio et al. 2009]. In curriculum learning, a model is trained on successively harder tasks until reaching the final task and was shown to yield improved performance on supervised classification tasks. In our case, the simpler game may be acting as such, helping the more complex one to find improved policies by providing guides for exploration of the policy space. As far as we know, this is the first observation of such phenomena in DRL.

Our main contributions can be summarized as follows. First, we show that A3C is able to learn at least two similar tasks simultaneously, without changes in the algorithm or network architecture. Second, we show that there it can be beneficial for individual tasks to be jointly trained with another. While this is only an initial take on this phenomena, these results are promising and can be equally useful for both multi-task learning and single-task learning.

Future works include experimenting with different tasks and different combinations, as well different number of tasks. The hypothesis proposed here, that simpler tasks help more complex tasks, must be thoroughly tested. It must also be observed if learning dissimilar tasks simultaneously do degrade performance as would be expected.

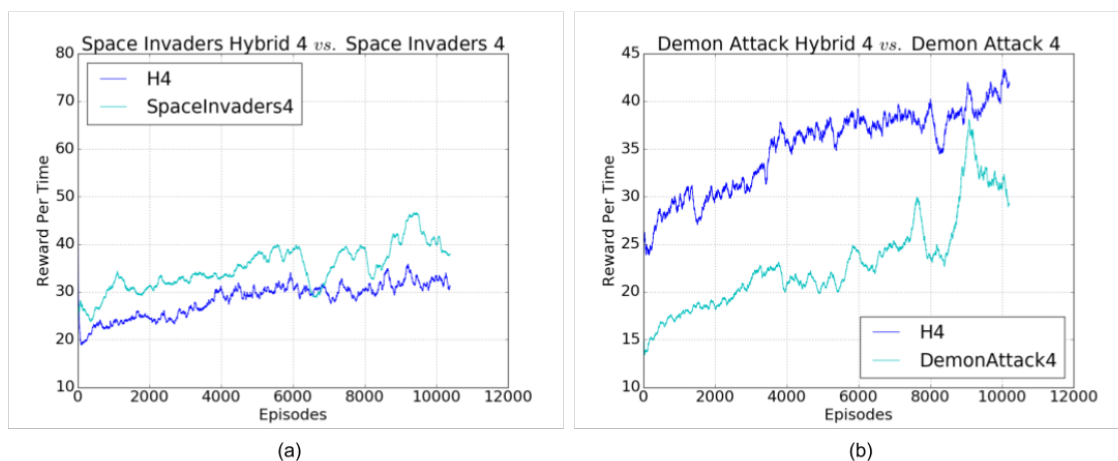


Figure 4. Results for standalone and joint training for (a) Space Invaders and (b) Demon Attack. H4 stands for the hybrid approach, with two threads being used for each game, while SpacelInvaders4 and DemonAttack4 stands for the standalone cases using four threads dedicated to sole game. Curves were smoothed using Exponential Weighted Moving Average in order to better observe general trends.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research. This work is supported by the Research Initiation Scholarship Program - Doctorate in Progress (PBIP-DA) from Federal University of Pelotas (UFPel).

References

- Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym.
- Buciluă, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM.
- Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A. A., Pritzel, A., and Wierstra, D. (2017). Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*.
- Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, page 201611835.

- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.
- Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
- Parisotto, E., Ba, J. L., and Salakhutdinov, R. (2015). Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*.
- Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., and Hadsell, R. (2015). Policy distillation. *arXiv preprint arXiv:1511.06295*.
- Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.
- Taylor, M. E. and Stone, P. (2011). An introduction to intertask transfer for reinforcement learning. *Ai Magazine*, 32(1):15.

Detecção de Cardiopatias por Eletrocardiograma Utilizando *Multilayer Perceptron*

Vinicius Suterio¹, Cristiano Marcos Agulhari², Danilo Hernane Spatti²

¹Centro Integrado de Pesquisa em Controle e Automação
Universidade Tecnológica Federal do Paraná (UTFPR)
Avenida Alberto Carazzai, 1640 CEP 86300-000 - Cornélio Procópio - PR - Brasil

²Departamento Acadêmico de Elétrica. Curso Eng. de Controle e Automação
Universidade Tecnológica Federal do Paraná (UTFPR) – Cornélio Procópio, PR – Brasil

vinicius.suterio@yahoo.com.br, {agulhari, danilospatti}@utfpr.edu.br

Abstract. *This paper consists in proposing of a system to aiding health's professional, aimed to facilitate the interpretation of patient's eletrocardiograms for detection of arrhythmia between ECG's patient healthy and with cardiopathy. The system uses the wavelet packet transform (TWP) to feature extraction of the ECGs and the data obtained after extraction of characteristics are presented for an algorithm, multilayer perceptron (MLP), automated. This algorithm was developed with the purpose of finding the best combination of parameters used by MLP for each of the chosen wavelet packet families, thus providing the best results. The project provided satisfactory results with a low degree of uncertainty.*

Keywords. *wavelet packet, multilayer perceptron, feature extraction, eletrocardiograms, cardiopathies.*

Resumo. *Este artigo consiste em propor um sistema para auxiliar o profissional da saúde, visando facilitar a interpretação do eletrocardiograma (ECG) de pacientes para a detecção de arritmias entre ECGs de pacientes saudáveis e com cardiopatia. O sistema utiliza da transformada wavelet packet (TWP) para a extração de características dos ECGs e os dados obtidos, após a extração de características, são apresentados para um algoritmo, multilayer perceptron (MLP), automatizado. Este algoritmo foi desenvolvido com o intuito de encontrar a melhor combinação dos parâmetros utilizados pela MLP para cada uma das famílias wavelet packet escolhidas fornecendo assim, resultados satisfatórios com baixo grau de incerteza.*

Palavras-chave. *wavelet packet, perceptron multicamadas, extração de características, eletrocardiogramas, cardiopatias.*

1. Introdução

Segundo o estudo realizado pela *World Health Organization* (WHO), publicado em [Organization 2014] e efetuado entre os anos de 2000 e 2012, a doença isquêmica cardíaca lidera a lista das dez principais causas de morte no mundo, atingindo 0,7 milhões de pessoas a mais do que o acidente vascular encefálico. No geral, as doenças cardiovasculares mataram 17,5 milhões de pessoas em 2012 (30% das mortes no total), sendo a doença isquêmica cardíaca responsável pela morte de 7,4 milhões de pessoas. No geral, aproximadamente 50% do total das mortes decorrem de algum tipo de cardiopatia. Apesar do avanço tecnológico no tratamento das cardiopatias, segundo informações da WHO, o número de casos de cardiopatias estão aumentando com o decorrer dos anos, havendo um acréscimo de 1,4 milhões em doze anos [Organization 2014]. Maiores informações, com dados estatísticos, podem ser encontradas em [Benjamin et al. 2017].

Frente à exposição dos dados apresentados, pode-se afirmar que a prevenção e diagnósticos de cardiopatias são essenciais para melhorar o atual quadro mundial. O processamento de eletrocardiogramas (ECGs) tem sido uma linha de pesquisa forte nas últimas décadas, como apresentado em [Monteiro and Farias 1985, Huallpa et al. 1992, Souza et al. 2006, Melo et al. 2017]. Um dos fatores que possibilitaram o fortalecimento de pesquisas nessa área foi a criação do banco de dados do *Massachusetts Institute of Technology* (MIT) no ano de 2001, o qual é composto por ECGs coletados em diversas situações e separados em várias categorias, que servem também como base de estudo [Goldberger et al. 2000].

Entre as diversas técnicas para detecção de padrões para o auxílio de diagnósticos, as Redes Neurais Artificiais (RNAs) têm se destacado na literatura. Tais técnicas em geral consistem na extração de características dos sinais em análise (utilizando, por exemplo, transformadas de Fourier ou *Wavelets*), e no posterior treinamento de uma RNA utilizando as classes de sinais a serem detectadas. Os trabalhos a seguir podem ser citados como exemplos da aplicação de RNAs.

Reconhecimento e classificação de cardiopatias utilizando ECGs pode ser encontrado em [Bueno 2006]. O trabalho possui foco na extração de características do ECG por intermédio do modelo matemático auto-regressivo (AR) e aplicação deles em uma rede neural artificial, identificando assim as cardiopatias.

Detecção de cardiopatia utilizando técnicas de otimização é apresentado em [Neto 2009]. Este trabalho apresenta técnicas de otimização por nuvem de partículas juntamente com a extração de características por intermédio de *wavelets*. Esses dados são aplicados em uma RNA para a detecção automática de arritmias cardíacas baseadas na análise de ECGs.

Desenvolvimento de um sistema de classificação de voz disfônica utilizando transformada *wavelet packet* e o algoritmo *Best Basis* (BBA) como redutor de dimensionalidade, por fim aplicação em RNAs, pode ser encontrado em [Crovato 2004].

Efetuar uma análise comparativa entre diferentes técnicas utilizadas para classificação dos movimentos dos dedos das mãos utilizando sinais mioelétricos, é apresentado em [Almeida and Faceroli 2014]. Os métodos de classificação utilizados neste trabalho foram dos k-vizinhos mais próximos (KNN) e a rede neural *perceptrons* de múltiplas camadas (MLP).

Este trabalho consiste em propor um processo de auxílio para o profissional da saúde, visando facilitar a interpretação do quadro do paciente para uma possível detecção de patologias. O método é baseado em duas etapas. Primeiramente, é realizado um processo de extração de características, dado pela aplicação da transformada *wavelet packet* em janelas de tamanho pré-determinado. Em seguida, os coeficientes de energia das folhas da transformada são utilizados como entrada em uma rede neural MLP, responsável por classificar o trecho do sinal entre um sinal saudável e um sinal com arritmia. Para determinar tanto a família *wavelet* e a quantidade de níveis de decomposições da transformada *wavelet packet*, bem como a estrutura da MLP mais apropriada, foi realizado um procedimento computacional automatizado. Os resultados obtidos ilustram um baixo grau de erro na classificação do sinal.

O restante deste artigo está organizado como segue. A Seção 2 apresenta os conceitos básicos sobre redes neurais artificiais (RNAs) com foco na RNA utilizada para o desenvolvimento deste trabalho, alguns conceitos de reconhecimento de padrões e extração de características. A Seção 3 são apresentados detalhes do desenvolvimento. A Seção 4 apresenta os resultados, Seção 5 são apresentadas as conclusões.

2. Fundamentação Teórica

O processamento de dados neste trabalho dar-se-á por intermédio da utilização de redes neurais artificiais. As RNAs são modelos computacionais inspirados no processo de aprendizagem do cérebro humano, emulando o seu funcionamento. Podem ser definidas por um conjunto de unidades de processamento, caracterizado por neurônios artificiais, interligados pelas sinapses artificiais que são representadas por vetores/matrizes de pesos sinápticos [Silva et al. 2016].

Dentre as diversas aplicações das RNAs, para este projeto, sua implementação estará restrita às aplicações para as seguintes utilidades, aproximador universal de funções, reconhecimento/classificação de padrões, agrupamento de dados (clusterização) e otimização de sistemas [Silva et al. 2016].

Dos modelos existentes para a configuração das redes neurais, o que mais se destaca é o *perceptron*, devido à sua capacidade em reconhecer padrões. Entre os tipos de redes *perceptron*, o que será utilizado neste trabalho será o *perceptron* multicamadas. Uma das suas principais vantagens é que uma vez pronta, não demanda tempo de processamento, apenas memória, o que a torna uma técnica melhor de implementação. Maiores informações sobre RNAs podem ser encontradas em [Silva et al. 2016, Kovács 2002].

Vale ressaltar que para o conjunto dos dados estudados seja processado ele não pode simplesmente ser aplicado na rede, há a necessidade de efetuar um pré-processamento. Tal processamento é chamado de normalização dos dados, e consiste em deixar os dados dentro do domínio da região dinâmica das funções de ativação, com o intuito de melhorar o desempenho do treinamento evitando a saturação na saída da RNA e problemas com o processamento dos dados. Em alguns casos há a necessidade até de um pós-processamento, chamado de desnormalização dos dados [Silva et al. 2016].

Para a utilização das RNAs é fundamental a seleção dos dados de entrada. Normalmente é realizada uma fase de reconhecimento de padrões e extração de características para tal. O reconhecimento de padrões é a ciência que atua no reconhecimento, descrição,

classificação e agrupamento de padrões. Trata-se da classificação e descrição de objetos e consiste em extrair características, selecioná-las e gerar um classificador referente ao objeto de estudo. Um sistema de reconhecimento de padrões envolve basicamente três aspectos: aquisição de dados e pré-processamento, a representação de dados e a tomada de decisão [Jain et al. 2000].

Dentre os tipos de abordagens para a execução de detecção de padrões, este projeto utilizará a abordagem neuronal que consiste na capacidade de aprender complexas relações não-lineares de entrada e saída. Sua principal característica é o reconhecimento de padrões de forma praticamente independente de conhecimento do domínio específico, uma vez que não há necessidade de saber do que se tratam os dados e grupos estudados [Jain et al. 2000]. Métodos de reconhecimento de padrões, aplicados na área de classificação de doenças cardiovasculares e utilizando também abordagem neuronal, pode ser encontrados em [Mattos et al. 2013].

O processo de extração de características pode ser chamado de compressão, uma vez que o sinal obtido possui menos informações que o sinal original, porém conservam-se as características mais relevantes do sinal estudado [Crovato 2004].

Neste trabalho será utilizada a ferramenta *wavelet* e os coeficientes serão obtidos pela sua decomposição usando a Transformada *wavelet packet*, que é uma generalização da decomposição *wavelet*. Os coeficientes da *wavelet packet* contém informações de energia dos sinais em cada banda de frequência, e isso é uma informação que auxilia na detecção dos padrões [Mallat 2008]. A escolha da função a ser utilizada depende da aplicação do sinal ou de propriedades que se desejam analisar [Agulhari 2009]. Maiores detalhes sobre *wavelet* e *wavelet packets* podem ser encontrados em [Mallat 2008].

3. Desenvolvimento

O fluxograma mostrado na Figura 1, apresenta a sequência do desenvolvimento do trabalho. Cada uma das atividades apresentadas são explanadas a seguir.

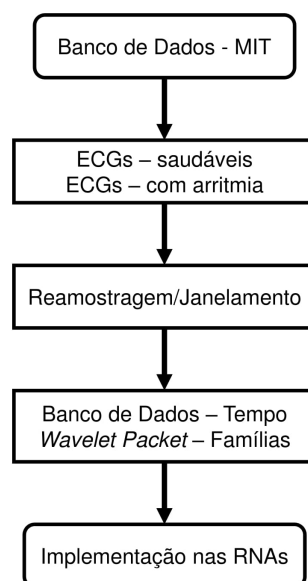


Figura 1. Fluxograma do desenvolvimento do projeto.

Os dados utilizados para o projeto foram adquiridos por intermédio do banco de dados do MIT [Goldberger et al. 2000]. Neste banco, podem-se encontrar sinais de ECG coletados de pessoas saudáveis e sinais de ECG compostos por cardiopatias como arritmias, doenças isquêmicas do coração, infarto do miocárdio, entre outras. Vale ressaltar que cada sinal vem acompanhado de metadados como o frequência de amostragem e duração total da coleta, que são necessários para o processamento [Goldberger et al. 2000].

A ferramenta computacional utilizada para a importação de dados, processamento e obtenção de resultados é o MATLAB® (*matrix laboratory*), que se trata de um *software* com grande capacidade de processamento normalmente utilizado no desenvolvimento de pesquisas científicas.

3.1. Banco de dados - MIT

Neste projeto utilizaram-se ECGs do banco de dados de pessoas saudáveis, composto por 90 ECGs (*The ECG-ID Database*), e com arritmia, composto por 48 ECGs (*MIT-BIH Arrhythmia Database*) [Goldberger et al. 2000]. Na Figura 2 são apresentados exemplos de ECGs saudáveis e com arritmia. É possível notar uma diferença visual entre os ECGs, porém a detecção e constatação da patologia no sinal não é trivial. Cada classe de sinal foi coletada utilizando diferentes taxas de amostragem. Para que o procedimento de extração de características ocorra de maneira normalizada, é necessário analisar a frequência de amostragem de cada ECG e fazer procedimentos de reamostragem para uniformizar tal característica, como detalhado na seção a seguir. A Tabela 1 apresenta o tempo de coleta e a frequência de amostragem para cada tipo de ECG.

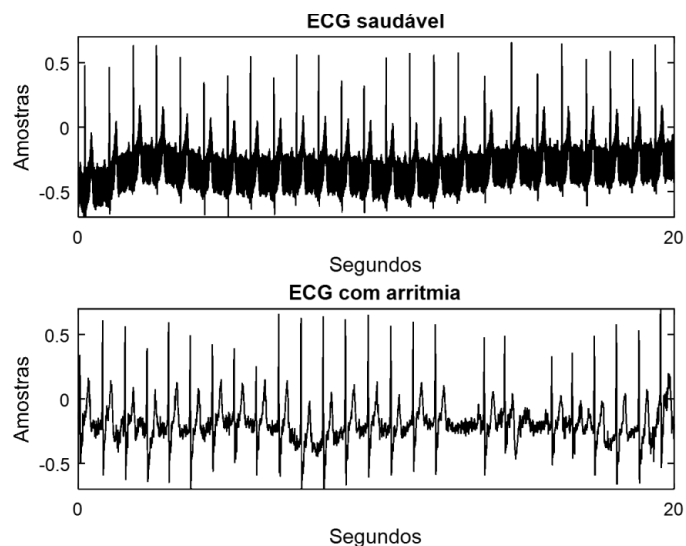


Figura 2. Exemplos ECG saudável e com arritmia.

Tabela 1. Dados fornecidos pelo MIT

Tipo de ECG	Tempo de coleta [s]	Frequência de amostragem [amostras/s]
Saudável	20	360
Arritmia	1856	500

3.1.1. Reamostragem e janelamento

Primeiramente, foi realizado o procedimento de reamostragem, utilizando o comando *resample* do MATLAB®, que é responsável por efetuar a reamostragem dados uniformes ou não a uma nova taxa de amostragem fixa, aplicando um filtro passa-baixa FIR *anti aliasing*. Foi escolhido reamostrar os ECGs para que os mesmos fossem representados a uma taxa de 360 amostras/segundo, por ser a mais baixa taxa de amostragem dada no banco de dados. De acordo com [Goldberger et al. 2000], os sinais coletados são originalmente limitados a 100 Hz, portanto tal taxa de amostragem satisfaz o critério de *Nyquist* (dobro da maior frequência) para não ocorrer *aliasing* - inconsistência de dados [Oppenheim et al. 1999].

Após a etapa de reamostragem, foi efetuado o janelamento dos sinais [Andrade and Soares 2013]. A Tabela 2 apresenta a quantidade de janelas e amostras obtidas para cada tipo de ECG. Utilizaram-se janelas de 1024 amostras, o que corresponde a aproximadamente 3 segundos, para a montagem do banco de dados final, conforme ilustrado na Figura 3.

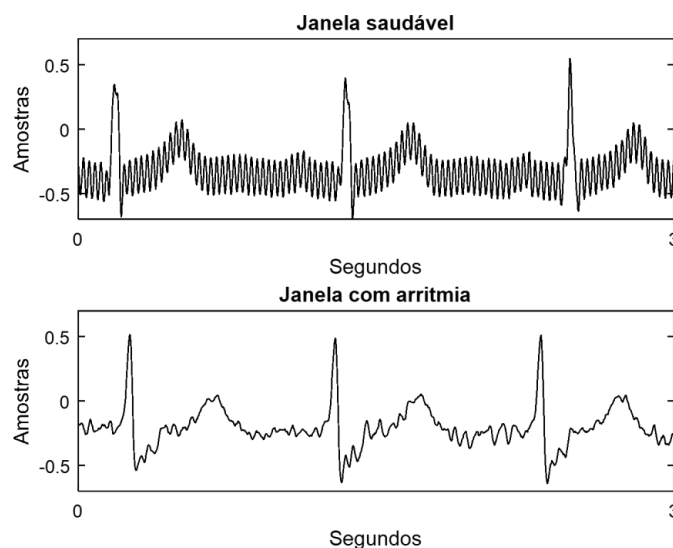


Figura 3. Exemplo de janelamento para ECGs saudáveis e com arritmia.

Tabela 2. Dados obtidos pelo janelamento

Tipo de ECG	Quantidade de janelas	Quantidade de amostras
Saudável	7	1024
Arritmia	634	

3.2. Wavelet Packets

A etapa de extração características considerada neste artigo consiste na aplicação da transformada *wavelet packet*. A energia de cada um dos conjuntos de coeficientes gerados pela TWP corresponde a uma entrada da rede neural MLP classificadora. Desta forma, a família *wavelet* utilizada para a transformada, bem como o número de

níveis de decomposição, afeta consideravelmente o desempenho da rede neural. Portanto, é necessário analisar qual combinação de família *wavelet* e número de níveis de decomposições é mais adequada para a aplicação. Para tal análise, foram escolhidos os níveis 3 e 4 de decomposição da árvore binária, resultando respectivamente em 8 e 16 folhas de energia. As famílias que foram escolhidas são apresentadas na Tabela 3.

Tabela 3. Famílias *wavelet packet*

Família <i>wavelet</i>	Ordem	Níveis
<i>Coiflets</i>	1 a 5	3 e 4
<i>Symlets</i>		
<i>Daubechies</i>	2 a 5	
<i>Haar</i>	-	

A família *Daubechies* foi escolhida por ser indicada como a melhor *wavelet* para comprimir sinais de ECG conforme [Besar et al. 2000], uma vez que se utiliza para efetuar a extração de características. Em contra partida, foram escolhidas mais 3 famílias de diferentes ordens para efetuar uma comparação dentre as famílias e suas respectivas ordens.

3.3. Banco de dados final

Primeiramente o conjunto de todas as janelas dos ECGs considerados foram organizados em uma única matriz de dimensão apresentada na Tabela 4. Após esta etapa, foi criado um banco de dados com todas as possíveis entradas para a rede possibilitando o processo automatizado de determinação da melhor estrutura da rede neural. Com isso, para cada família e cada escolha de níveis de decomposição, foi criado um arquivo com as energias de todas as folhas da *wavelet packet* apresentados respectivamente nas Tabelas 5 e 6.

O banco de dados final total foi composto por 30 bancos distintos obtidos de todas as famílias com suas respectivas ordens, sendo 15 bancos do nível 3 e os outros 15 do nível 4.

Tabela 4. Conjunto de todas as janelas dos ECGs

Tipo de ECG	Quantidade de janelas	Dimensão da matriz
Saudável	630	31062x1024
Arritmia	30432	

Tabela 5. Conjunto de ECGs *wavelet packet* para o nível 3

<i>wavelet packet</i>	Quantidade de banco de dados gerados	Dimensão da matriz
<i>Coiflets</i>	5 banco de dados distintos	31062x8
<i>Symlets</i>	5 banco de dados distintos	
<i>Daubechies</i>	4 banco de dados distintos	
<i>Haar</i>	1 banco de dados	

Tabela 6. Conjunto de ECGs *wavelet packet* para o nível 4

<i>wavelet packet</i>	Quantidade de banco de dados gerados	Dimensão da matriz
<i>Coiflets</i>	5 banco de dados distintos	31062x16
<i>Symlets</i>	5 banco de dados distintos	
<i>Daubechies</i>	4 banco de dados distintos	
<i>Haar</i>	1 banco de dados	

3.4. Procedimento para determinação da estrutura da RNA

Os dados serão aplicados a uma rede neural artificial, *perceptron* multicamadas, com uma camada neural escondida. Este tipo de rede foi escolhida devido à sua capacidade em reconhecer padrões e também foi utilizado em [Bueno 2006] para uma classificação de cardiopatias.

A RNA possui treinamento supervisionado, que consiste em apresentar à rede qual seria o resultado desejado. Assim, para cada janela de cada um dos ECGs foi acrescentada uma coluna que codifica a classificação do sinal, sendo 1 se o ECG for saudável e 0 se for um sinal com arritmia. Tanto para o nível três como o quatro, de todas as famílias *wavelet packet*, gerou-se uma matriz composta pela concatenação das energias de cada uma das janelas obtidas acrescentando a coluna que codifica a classificação do sinal. Com isso, a matriz final obtida para cada um dos níveis apresenta as dimensões conforme a Tabela 7. Dos dados utilizados, 60% dos foram para treinamento e 40% para teste.

Tabela 7. Matriz para implementação nas RNAs

Nível	Dimensão da matriz final
3	31062x9
4	31062x17

Desenvolveu-se um algoritmo automatizado para encontrar a melhor configuração da RNA para cada família, considerando todas as combinações dos parâmetros apresentados na Tabela 8. O algoritmo consiste em, para cada combinação de parâmetros, treinar uma RNA e armazenar para a melhor delas o menor erro obtido, a porcentagem de classificação corretas, a quantidade de neurônios utilizada, o erro estipulado para a rede e a taxa de aprendizagem. Os resultados serão apresentados na seção seguinte.

Tabela 8. Combinação dos parâmetros para a RNA automatizada

-	Quantidade de neurônios	Erro	Aprendizagem
Mínimo e máximo	5 a 30	1×10^{-1} a 1×10^{-6}	0,1 a 0,5
Variação	5 em 5	1×10^{-1} em 1×10^{-1}	0,1 em 0,1

Após encontrar a melhor configuração de RNA para cada uma das famílias *wavelet packet*, efetuaram-se dez treinamentos para verificar a capacidade de generalização da rede e conseqüentemente obter o menor erro possível.

4. Resultados

Após executar o procedimento detalhado na Seção 3.4, determinou-se que a melhor configuração obtida, resultou em 99,9% de acertos, corresponde à utilizada da família

Daubechies 3, com 4 níveis de decomposição, para a fase de extração de características, e subsequente aplicação em uma rede neural com 5 neurônios na camada escondida, erro estipulado para a rede de 1×10^{-5} e taxa de aprendizagem de 0,1. As Tabelas 9 e 10 apresentam os resultados para todas as combinações de parâmetros consideradas.

Tabela 9. Família *wavelet* nível 3.

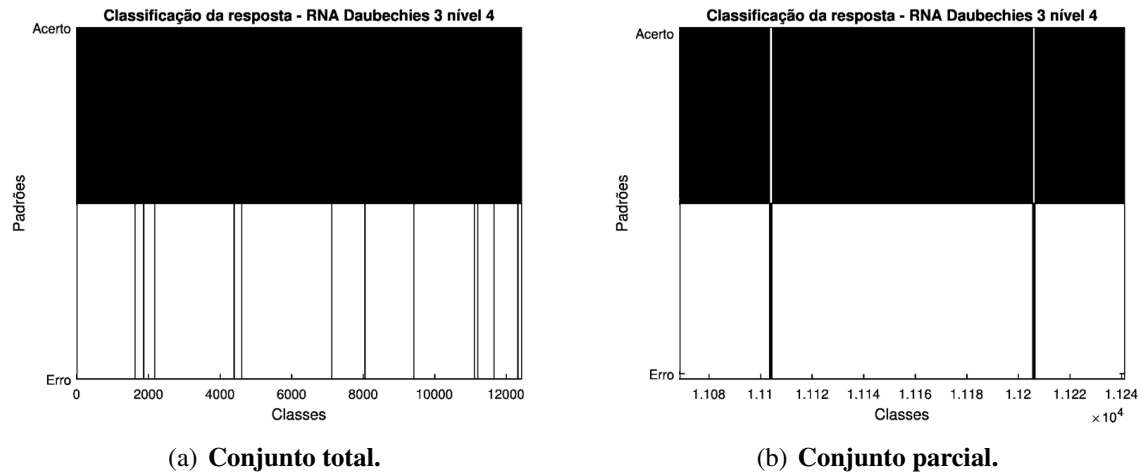
Família <i>Wavelet</i> Nível 3	Acerto	Menor Erro	Neurônio	Erro	Taxa
<i>Coiflets 1</i>	99,0221%	370	5	1×10^{-4}	0,4
<i>Coiflets 2</i>	99,8390%	20	5	1×10^{-4}	0,4
<i>Coiflets 3</i>	98,6720%	165	5	1×10^{-6}	0,1
<i>Coiflets 4</i>	99,7746%	28	5	1×10^{-5}	0,2
<i>Coiflets 5</i>	99,8471%	19	5	1×10^{-4}	0,2
<i>Symlets 1</i>	93,9477%	752	15	1×10^{-5}	0,2
<i>Symlets 2</i>	99,2998%	87	5	1×10^{-3}	0,2
<i>Symlets 3</i>	97,3924%	324	5	1×10^{-5}	0,3
<i>Symlets 4</i>	97,3119%	334	5	1×10^{-6}	0,3
<i>Symlets 5</i>	98,3501%	205	5	1×10^{-6}	0,2
<i>Daubechies 2</i>	96,9980%	373	5	1×10^{-6}	0,3
<i>Daubechies 3</i>	95,8793%	512	5	1×10^{-4}	0,4
<i>Daubechies 4</i>	95,6781%	537	5	1×10^{-6}	0,3
<i>Daubechies 5</i>	97,8270%	270	5	1×10^{-4}	0,5
<i>Haar</i>	94,8008%	646	5	1×10^{-4}	0,1

Tabela 10. Família *wavelet* nível 4.

Família <i>Wavelet</i> Nível 4	Acerto	Menor Erro	Neurônio	Erro	Taxa
<i>Coiflets 1</i>	99,8390%	20	5	1×10^{-4}	0,1
<i>Coiflets 2</i>	98,4064%	198	10	1×10^{-5}	0,2
<i>Coiflets 3</i>	99,5091%	61	5	1×10^{-6}	0,1
<i>Coiflets 4</i>	99,3964%	75	5	1×10^{-4}	0,1
<i>Coiflets 5</i>	99,5976%	50	5	1×10^{-5}	0,1
<i>Symlets 1</i>	99,7666%	29	5	1×10^{-4}	0,5
<i>Symlets 2</i>	99,8712%	16	5	1×10^{-5}	0,5
<i>Symlets 3</i>	99,8149%	23	5	1×10^{-6}	0,5
<i>Symlets 4</i>	99,3400%	82	5	1×10^{-5}	0,2
<i>Symlets 5</i>	99,6217%	47	5	1×10^{-6}	0,4
<i>Daubechies 2</i>	99,8310%	21	5	1×10^{-6}	0,2
<i>Daubechies 3</i>	99,9034%	12	5	1×10^{-5}	0,1
<i>Daubechies 4</i>	99,6949%	38	5	1×10^{-4}	0,4
<i>Daubechies 5</i>	98,4628%	191	5	1×10^{-6}	0,2
<i>Haar</i>	98,4064%	198	15	1×10^{-4}	0,3

É importante ressaltar que, para a maioria dos casos, as redes com 5 neurônios na camada escondida foram as que resultaram as melhores classificações, sendo considerado um resultado positivo dado que tal configuração implica que redes de menor complexidade são capazes de resolver o problema proposto.

As Figuras 4 (a) e 4 (b) mostram a quantidade de acertos e erros do conjunto total de amostras e do conjunto parcial de amostras, respectivamente, para a melhor configuração obtida.



(a) Conjunto total.

(b) Conjunto parcial.

Figura 4. Classificação da resposta - RNA *Daubechies* 3 nível 4.

A Figura 5 apresenta a matriz de confusão de dimensão 2x2 que foi obtida a partir da análise do vetor de saída, fornecido pela rede neural artificial, utilizando a melhor combinação de parâmetros.

Na área da medicina, a tomada de decisão deve ser extremamente precisa e por isso, neste projeto, adotou-se uma margem de tolerância de 0,01. Com isso os dados estudados só podem ser considerados 1 apenas se forem maior ou igual à 0,99 e devem ser considerados 0 apenas se forem menor ou igual à 0,01. Então a margem de 0,01 à 0,99 é considerada como faixa de indeterminação, onde existem dados que deveriam ser 0 e foram considerados como 1 e dados que deveriam ser 1 e foram considerados como 0.

MATRIZ DE CONFUSÃO		
	0	1
0	12158	7
1	5	255

Figura 5. Matriz de confusão do vetor de saída da rede, referente à *Daubechies* 3 nível 4.

Na matriz de confusão, o elemento (i,j) corresponde à quantidade de dados que pertencem à classe i, porém foram classificados como pertencentes à classe j pelo método proposto. Assim, quanto maior a quantidade de elementos classificados na diagonal principal da matriz de confusão, melhor é a resposta obtida. No projeto, significa que a RNA obteve grande êxito ao classificar os padrões entre 0 e 1. De todos os 12425 dados para testes, referente aos 40%, 12413 foram classificados corretamente.

Quanto maior a quantidade de elementos classificados na diagonal principal da matriz de confusão, melhor é a resposta obtida. No presente trabalho, significa que a RNA obteve grande êxito ao classificar os padrões entre 0 e 1. De todos os 12425 dados para testes, referente aos 40%, 12413 foram classificados corretamente.

5. Conclusões

Neste trabalho, foi proposta uma técnica de auxílio para o profissional da saúde, visando facilitar a interpretação do eletrocardiograma do paciente para uma possível detecção de cardiopatias. O projeto, propôs uma técnica de detecção automática de cardiopatias a partir de sinais de eletrocardiogramas com o intuito de gerar resultados com baixo grau de incerteza.

O método consiste na aplicação de uma fase de extração de características em um conjunto de ECGs de pacientes saudáveis e com arritmia por intermédio de (TWP) e sua subsequente aplicação em um algoritmo automatizado de uma rede neural (MLP). Após a aplicação do procedimento automatizado, implementado para comparar os resultados de diferentes combinações de parâmetros da TWP e da rede neural, pode-se dizer que para detectar ECGs saudáveis e com arritmia, a melhor *wavelet packet* para ser utilizada é da família *Daubechies*, mais especificamente *Daubechies 3* nível 4 - 16 folhas de energia. Com configuração da rede neural composta por 5 neurônios na camada escondida, erro 1×10^{-5} e taxa de aprendizagem de 0,1. Tal combinação resultou em 99,9% de acertos na classificação de ECGs entre saudáveis e com arritmia, o que é um resultado considerado satisfatório para o auxílio de decisões de profissionais da saúde.

Referências

- Agulhari, C. M. (2009). Compressão de eletrocardiogramas usando wavelets.
- Almeida, R. and Faceroli, S. (2014). Análise comparativa das técnicas knn e rede neural mlp na classificação de padrões mioelétricos.
- Andrade, A. O. and Soares, A. B. (2013). Técnicas de janelamento de sinais. Access date: 15 abr. 2017.
- Benjamin, E. J., Blaha, M. J., Chiuve, S. E., Cushman, M., Das, S. R., Deo, R., de Ferranti, S. D., Floyd, J., Fornage, M., Gillespie, C., Isasi, C. R., Jiménez, M. C., Jordan, L. C., Judd, S. E., Lackland, D., Lichtman, J. H., Lisabeth, L., Liu, S., Longenecker, C. T., Mackey, R. H., Matsushita, K., Mozaffarian, D., Mussolino, M. E., Nasir, K., Neumar, R. W., Palaniappan, L., Pandey, D. K., Thiagarajan, R. R., Reeves, M. J., Ritchey, M., Rodriguez, C. J., Roth, G. A., Rosamond, W. D., Sasson, C., Towfighi, A., Tsao, C. W., Turner, M. B., Virani, S. S., Voeks, J. H., Willey, J. Z., Wilkins, J. T., Wu, J. H., Alger, H. M., Wong, S. S., Muntner, P., and (2017). Heart disease and stroke statistics—2017 update: A report from the american heart association. *Circulation*.
- Besar, R., Eswaran, C., Sahib, S., and Simpson, R. J. (2000). On the choice of the wavelets for ecg data compression. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 6, pages 3614–3617 vol.6.
- Bueno, N. M. (2006). Classificação automática de cardiopatias baseada em eletrocardiograma.

- Crovato, C. D. P. (2004). Classificação de sinais de voz utilizando a transformada *wavelet packet* e redes neurais artificiais.
- Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23):e215–e220.
- Huallpa, B. N., Costa, E. T., and Lopes, A. (1992). Obtenção e processamento de sinais de eletrocardiografia de alta resolução. Access date: 07 jun. 2017.
- Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37.
- Kovács, Z. L. (2002). *Redes neurais artificiais*. Editora Livraria da Física.
- Mallat, S. (2008). *A wavelet tour of signal processing: the sparse way*. Academic press.
- Mattos, C. L., Júnior, A. H. S., Neto, A. R. R., Barreto, G. A., Ramos, R. F., Mazza, H. A., and Mota, M. O. (2013). Cardiovascular disease risk classification from health insurance: A big data experimental comparison.
- Melo, S., Silva, H., and Fontes, A. (2017). Redução de ruídos no sinal de ecg usando um filtro passa alta. Access date: 07 jun. 2017.
- Monteiro, A. O. and Farias, M. A. C. (1985). Monitor inteligente de arritmias cardíacas. Access date: 07 jun. 2017.
- Neto, A. B. (2009). Detecção automática de arritmia cardíaca utilizando *wavelets* e redes neurais treinadas com otimização por nuvem de partículas.
- Oppenheim, A. V., Schafer, R. W., and Buck, J. R. (1999). *Discrete-time signal processing*. Pearson Education India.
- Organization, W. H. (2014). The top 10 causes of death. Access date: 5 ago. 2016.
- Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B., and dos Reis Alves, S. F. (2016). *Artificial Neural Networks - A practical course*. Springer.
- Souza, C. B., Andreão, R. V., and Segatto, M. V. (2006). Processamento de sinais de ecg para geração automática de alarmes. Access date: 07 jun. 2017.

CNNFusion-Gender: metodologia de fusão de partes faciais para a classificação de gênero em ambientes não controlados

Flávio Henrique da Silva^{1 2}

¹dod -Smart X, Curitiba, Brasil

²Universidade Federal do Paraná

{flavio}@executive.com.br, {flaviosilva}@ufpr.br

Abstract. *This work presents a gender classification algorithm using only facial parts (eyes, nose and mouth). The presented method uses a Convolutional Neural Network (CNN) architecture. The proposed method introduce a new CNN architecture called CNNFusion-Gender. The CNNFusion-Gender core is composed by three CNN's running in parallel with main goal of extract characteristics from the facial parts. Finally, employing the Fusion technique between the intermediate layers of the CNNFusion-Gender the method classify the persons gender. The approach was applied in two large publicly available datasets: CelebA and Faces of the World. The achieved results are competing against the state-of-the-art.*

Resumo. *O trabalho apresenta um algoritmo de classificação de gênero utilizando apenas partes faciais (olhos, nariz e boca). O método apresentado é composto por uma arquitetura de Rede Neural Convolutacional (CNN) chamado de CNNFusion-Gender. O núcleo do CNNFusion-Gender é composto por três CNN's sendo executadas em paralelo com o objetivo principal de extrair características a partir de partes faciais. Utilizando a técnica da fusão entre as camadas intermediárias da CNNFusion-Gender o método é capaz de classificar o gênero de indivíduos. A abordagem foi aplicada em duas grandes bases de dados públicas: CelebA e Faces of the World. Os resultados alcançados são competitivos com trabalhos estado-da-arte.*

1. Introdução

Atributos e partes faciais são utilizados em várias aplicações, tais como reconhecimento facial [Kumar et al. 2009, Song et al. 2014], identificação [Manyam et al. 2011] e classificação de gênero [Kumar et al. 2009, Li et al. 2012, Liu et al. 2015]. Considerando tais aplicações, a detecção destes atributos e partes faciais a partir de imagens sem restrições (*in the wild*) continuam sendo um desafio, principalmente, por causa da variação de pose, iluminação e oclusão presentes nos conjuntos de imagens.

No entanto, como tentativa de solucionar os desafios relacionados ao processamento de imagens *in the wild*, abordagens que empregam *landmarks* foram utilizadas no processo de detecção automática das partes faciais [Zhu and Ramanan 2012, Belhumeur et al. 2013, Smith et al. 2013]. Contudo estes métodos apresentam limitações em relação a dependência da estimativa eficaz das *landmarks* para o processo de detecção das regiões da face. Assim, com o objetivo de compensar as limitações apresentadas com

a utilização de *landmarks*, utilizou-se o método considerado estado da arte na detecção de objetos genéricos proposto por Ren *et al.*, denominado Faster R-CNN [Ren et al. 2015], totalmente livre de *landmarks*, onde foi possível obter resultados satisfatórios em relação a etapa de detecção das partes faciais.

Na literatura, recentes trabalhos discutem abordagens específicas relacionadas à análise facial, dentre elas a classificação de gênero em imagens *in the wild* [Ehrlich et al. 2016, Ranjan et al. 2016b, Zhang et al. 2014, Wang et al. 2016, Ranjan et al. 2016a]. Nesse sentido, Liu *et al.* [Liu et al. 2015] propuseram um método de predição de atributos faciais em imagens sem restrições que visa o treinamento de duas CNNs simultaneamente, sendo uma responsável pela detecção da face e, a outra desempenha a função de localização e predição dos atributos (por exemplo, olhos pequenos, boca larga, óculos de sol, cabelo). O processo de localização dos atributos faciais está totalmente associado a detecção da face, uma vez que a mesma não seja detectada corretamente a arquitetura responsável pelos atributos apresenta dificuldades em determinar corretamente a quantidade de atributos presentes em uma determinada imagem e suas respectivas localizações, dificultando, assim, o processo de classificação de gênero.

Nesse sentido, Zhang *et al.* [Zhang et al. 2014] apresentam uma nova arquitetura de CNN que realiza a representação da imagem de entrada em partes e, ao mesmo tempo, possibilita o alinhamento destas partes facilitando o processo de classificação dos atributos. Esta abordagem permite analisar simultaneamente as partes da imagem e suas respectivas poses individualmente, ao passo que extrai as características das mesmas. O diferencial deste método está em que cada CNN é capaz de aprender uma característica específica para uma determinada pose. A classificação final dos atributos é realizada por meio de um classificador SVM linear, que permite unir todas as características extraídas em relação a cada pose, proporcionando assim, a combinação de atributos (saia e cabelos longos) para realização de outras tarefas, como a classificação de gênero.

Recentemente, novos métodos que utilizam a abordagem de multitarefa estão sendo utilizados, proporcionando um melhoramento em relação a classificação de gênero. Ranjan *et al.* [Ranjan et al. 2016a] propuseram o HyperFace que executa simultaneamente várias tarefas como: detecção da face, localização das *landmarks*, estimativa da pose e reconhecimento de gênero. O método enfatiza a importância do conceito de multitarefa em CNNs e da técnica de fusão entre camadas intermediárias da arquitetura de rede proposta. Em seu trabalho mais recente, Ranjan *et al.* [Ranjan et al. 2016b] abordam novamente o conceito de multitarefa, propondo um novo framework de aprendizagem de multitarefas que permite a regularização dos parâmetros compartilhados pela CNN. Similar ao HyperFace [Ranjan et al. 2016a], também é aplicada a técnica de fusão entre a primeira, terceira e quinta camada da rede, permitindo que o treinamento das tarefas seja feito de forma independente em relação as demais. O processo de fusão entre as camadas é aplicado, pois, uma vez que uma arquitetura de uma CNN apresenta uma grande quantidade de *feature maps*, a dimensão do vetor de características é muito maior em relação a uma arquitetura com poucas camadas, impossibilitando que o processo de aprendizagem de todas as tarefas seja eficiente. Além disso, o processo de fusão visa combinar as características de um subespaço através de uma combinação linear ou não linear, atingindo um alto índice de acerto na etapa de classificação de uma determinada aplicação, por exemplo, gênero.

Em Ehrlich *et al.* [Ehrlich et al. 2016] apresentam uma nova abordagem de aprendizagem para classificação de vários atributos faciais. A representação das características é feita por meio de Restricted Boltzmann Machine (RBM) [Hinton et al. 2006]. A etapa inicial do método consiste na detecção das *landmarks* da face e da Principal Components Analysis (PCA). Após a obtenção das *landmarks* e PCA, os dados são enviados para o modelo Multi-Task Restricted Boltzmann Machine (MT-RBM) gerado com o auxílio da RBM para classificação dos atributos. Uma desvantagem do método está relacionado com a geração do modelo de classificação, pois caso a estimativa das *landmarks* seja realizada incorretamente o modelo apresentará inconsistência no processo de aprendizagem dos atributos.

Wang *et al.* [Wang et al. 2016] propuseram um novo método de aprendizagem relacionado a representação e classificação de atributos faciais através da utilização de informações de geolocalização, meteorológicas e contextuais de um determinado vídeo, que permitem determinar certos atributos de um indivíduo sem a necessidade de uma anotação manual das características do mesmo. A vantagem deste método está presente no processo de treinamento que não requer anotações de todos os atributos presentes no vídeo, o processo de aprendizagem encarrega-se de aprender automaticamente a representação de tais atributos através da análise das informações contextuais presentes na cena.

Liao *et al.* [Liao et al. 2017] propõem uma arquitetura de CNN simplificada, para o processo de estimativa de idade e classificação de gênero através da extração das características de 9 (nove) partes diferentes da face. Estas partes abrangem as regiões dos olhos, nariz e boca, sendo que a saída final corresponde o cálculo da média das predições referente a cada classe em relação as nove partes recortas da face. Uma limitação apresentada no método proposto está correlacionado a utilização do método de frontalização nas imagens face, aplicando assim, a abordagem em apenas em imagens frontais.

Considerando as informações destacadas, as principais contribuições apresentadas neste trabalho são: a) uma abordagem automática, livre de *landmarks* para classificação de gênero; b) uma nova arquitetura de CNN, que permite a fusão das características das partes faciais, sem que haja a necessidade da predição de atributos faciais para resolver o problema de classificação de gênero; c) treinamento de um único classificador, capaz de classificar um gênero apenas com a análise de uma parte específica da face; d) resultados competitivos, robustos e favoráveis com o estado da arte.

O restante deste trabalho está organizado da seguinte forma: a seção 2 descreve a nossa abordagem e a arquitetura de rede utilizada para realização dos experimentos de classificação de gênero, juntamente com a especificação da composição da arquitetura, na seção 3 contém uma breve explicação do funcionamento do detector utilizado, além das bases de dados utilizadas para a realização dos experimentos e, os respectivos resultados obtidos com a abordagem proposta. Por fim, a seção 4 mostra uma breve discussão em relação aos resultados obtidos.

2. CNNFusion-Gender

As *Convolutional Neural Networks* estabeleceram, recentemente, o estado da arte em diversas áreas da visão computacional [Taigman et al. 2014, Girshick 2015, Russakovsky et al. 2015, Hu et al. 2015, Levi and Hassner 2015], inclusive na solução de

problemas relacionados a classificação de gênero [Zhang et al. 2014, Wang et al. 2016, Ranjan et al. 2016a, Liu et al. 2015, Ehrlich et al. 2016, Ranjan et al. 2016b]. Estes demonstraram-se eficientes na solução de problemas de classificação que exigem um número limitado de classes (por exemplo, 2 classes), além disso outra vantagem presente nas CNNs está em relação a independência da necessidade de ter que definir explicitamente os descritores utilizados para extração das características.

Portanto, com os benefícios apresentados pelas CNNs e com objetivo de propor uma abordagem completamente automática, aplicou-se uma CNN de detecção automática para face e suas respectivas regiões (olhos, boca e nariz). Para esta finalidade, aplicou-se o detector denominado Faster R-CNN [Ren et al. 2015] a fim de proporcionar um processo de detecção mais eficaz.

A arquitetura proposta denominada CNNFusion-Gender é composta por três CNNs. Cada uma possuem três camadas convolucionais, uma camada de concatenação (fusão), responsável pela combinação das características, seguida por três camadas totalmente conectadas. As *Rectified Linear Units* (ReLU) [Nair and Hinton 2010] são aplicadas após cada camada convolucional. Nesta camada, aplica-se funções de ativação de não saturação, essas funções aumentam a não linearidade da função de decisão da rede, sem afetar os campos receptivos e, em seguida uma camada de *pool* responsável pela operação de *down-sampling* e uma camada de normalização estão presentes após a primeira e a segunda camada convolucional. A CNNFusion-Gender consiste em uma adaptação da rede proposta por Zavan *et al.* [Zavan et al.], que tem como objetivo a estimativa da pose por meio do nariz.

Por último, aplicou-se a camada de *Loss*, responsável por direcionar o aprendizado. Esta camada analisa, durante o treinamento, o desvio entre a saída e como as informações de entrada foram classificadas, processo representado pela expressão do erro da classificação, cujo objetivo é a maior minimização possível. Existem diversas funções de *Loss* entre as mais utilizadas estão: (i) *Softmax*, (ii) *Sigmoidal* com cruzamento de entropia e (iii) euclidiana. A função *Softmax* é utilizada para separar uma classe em relação a outra. A *Sigmoidal*, juntamente com cruzamento de entropia, visa prever os valores probabilísticos de 0 a 1. No entanto, a função euclidiana prevê valores infinitos e reais.

Em nossa abordagem, aplicou-se a *cross-entropy loss* expressa pela equação (1), para pudesse ser medido o erro em uma camada *Softmax*. O mesmo processo foi aplicado nos trabalhos propostos por Rajan *et al.* [Ranjan et al. 2016b, Ranjan et al. 2016a].

$$L_G = -(1 - g).log(1 - p_g) - g.log(p_g) \quad (1)$$

onde o $g = 0$ quando for masculino e 1 para feminino, p_g é a probabilidade de que o atributo de entrada seja do gênero masculino ou feminino.

Para o processo de classificação de gênero, executou-se três CNNs paralelamente, todas com a mesma arquitetura, cada uma corresponde a uma parte facial. A quarta camada encarrega-se de efetuar concatenação (fusão) das características extraídas das partes faciais (Figura 1), aumentando assim a eficiência do classificador treinado. Além desta arquitetura proposta que permite a fusão, utilizamos a arquitetura de Zavan *et al.* [Zavan et al.], onde alterou-se as informações de entrada, o que possibilitou que a rede

recebesse apenas imagens das regiões faciais. Além disso, a camada de fusão não foi adicionada a rede, com o objetivo de realizar o treinamento individualmente de cada parte, visando assim uma comparação com o método.

O Algoritmo 1 descreve, passo a passo, o processo de classificação de gênero. O primeiro passo visa aplicação da Faster R-CNN [Ren et al. 2015] para detecção da face, em seguida, a região correspondente a face é cortada e, novamente, a Faster R-CNN é aplicada sobre a face detectada para que as partes faciais (olhos nariz e boca) sejam detectados. No próximo passo, os tamanhos das partes são normalizados para 56×56 pixels, dimensão definida através dos bons resultados apresentados na Helen dataset [Le et al. 2012]. Em seguida, realiza-se a extração das características das imagens, processo realizado pelas camadas convolucionais, que recebem como entrada a saída da camada anterior, sendo que a única camada que não recebe informações das camadas anteriores é a primeira camada que tem como entrada as regiões dos olhos, boca e nariz. E para conclusão do processo aplica-se a técnica de fusão das características extraídas nas três CNNs e, após a concatenação destas características, todas são enviadas para as camadas totalmente conectadas, gerando assim a classificação final das imagens.

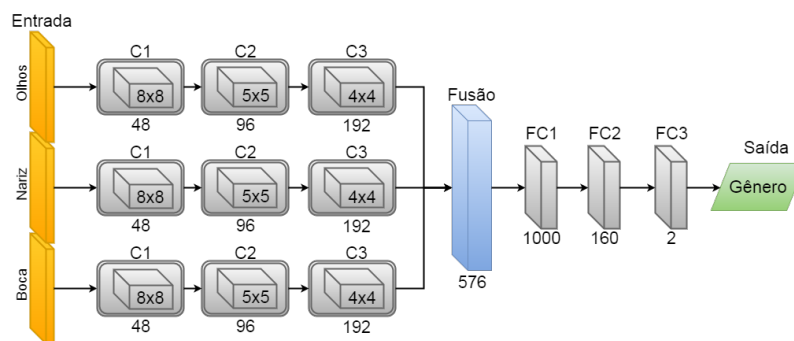


Figura 1. Arquitetura da CNNFusion-Gender

Algorithm 1 Algoritmo de Classificação de Gênero

```

function CLASSIFICACAODEGENERO(img)
  Recortar a img na face
  Recortar a img nos olhos, nariz e boca
  Normalizar img
  Carregar arquitetura da rede
  Carregar modelo treinado
  Inicializar rede com modelo treinado
  Colocar a rede no modo de teste
  Fornecer img como entrada à rede
  Realizar fusão das características extraídas na conv3
  Executar a propagação para frente do backpropagation
  Executar a cross-entropy
  Classifica o resultado como label
  Mapeia o label a um gênero
  return gênero
end function

```

O resumo de todo processo é ilustrado na Figura 2, desde a aquisição das imagens, passando pela detecção da face e das partes faciais. Na sequência, há aplicação da CNNFusion-gender até a classificação das imagens;

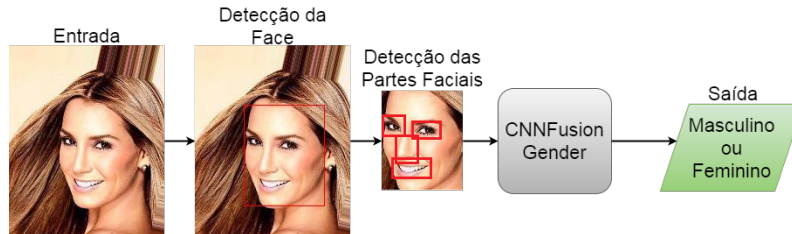


Figura 2. Visão geral do processo de classificação de gênero.

3. Experimentos

3.1. Detecção das partes faciais

A Faster R-CNN [Ren et al. 2015] é o estado-da-arte na detecção de objetos. Através da *Region Proposal Networks (RPN)*, são geradas regiões candidatas a partir da imagem de entrada da face e das partes faciais. As regiões são avaliadas através do *score* de confiabilidade fornecido pela métrica *Intersection over-Union (IoU)* [Ren et al. 2015], que permite calcular a interseção simétrica entre o *ground-truth* da região correspondente ao atributo com as coordenadas dos *bounding boxes* gerados das regiões candidatas, selecionando assim as melhores regiões para o treinamento.

Para avaliar individualmente o desempenho da Faster R-CNN em relação a detecção das partes faciais, utilizou-se o conjunto de dados Helen [Le et al. 2012], CelebA dataset [Liu et al. 2015] e FotW dataset [Escalera et al. 2016]. A métrica do coeficiente de intersecção proposto por Hoover *et al.* [Hoover et al. 1996] foi aplicada para avaliar a taxa de acurácia, resultados são mostrados na Tabela 1. Em relação ao treinamento do detector da face, utilizou-se as mesmas bases de dados utilizadas para treinar os detectores das partes faciais.

A Tabela 1 exibe a taxa de detecção das partes faciais e da face referente a cada conjunto de dados utilizado no processo de classificação de gênero. A ordem de classificação dos métodos estão conforme o valor da métrica *F-score* [Goutte and Gaussier 2005, Sokolova and Lapalme 2009].

Tabela 1. Taxa de acurácia de detecção das partes faciais e da face

Conjunto de dados	Olhos	Nariz	Boca	Face
CelebA [Liu et al. 2015]	90.38%	93.71%	92.88%	96.60%
FotW [Escalera et al. 2016]	85.39%	91.10%	89.63%	90%

Com o objetivo de consolidar a abordagem proposta, um comparativo foi realizado com o método utilizado na detecção das partes faciais [Ren et al. 2015] no conjunto de dados Helen com demais métodos estado-da-arte, com vistas a detecção das partes faciais, tais: como olhos, nariz e boca. A Tabela 2, exibe os resultados que estão organizados em ordem crescente, conforme a taxa de detecção das partes.

Tabela 2. Resultados comparativos da detecção de atributos faciais na base de dados Helen

Métodos	Olhos	Nariz	Boca
Zhu e Ramanan [Zhu and Ramanan 2012]	53.30%	<i>n/a</i>	68.70%
Saragih <i>et al.</i> [Saragih et al. 2009]	67.90%	89%	76.90%
Liu <i>et al.</i> [Liu et al. 2011]	77%	84.30%	74.20%
Gu e Kanade [Gu and Kanade 2008]	74.30%	88.90%	78.90%
Smith <i>et al.</i> [Smith et al. 2013]	78.50%	92.20%	85.70%
Faster R-CNN [Ren et al. 2015]	77.40%	93.47%	86.23%

Em geral, vemos que a Faster R-CNN [Ren et al. 2015] compara-se favoravelmente com quase todos os trabalhos anteriores considerados estado-da-arte sobre este conjunto de dados, exceto quando comparado com a detecção dos olhos com a abordagem de Smith *et al.* [Smith et al. 2013]. Os métodos comparados visam a utilização das partes faciais para o alinhamento e reconstrução da face, objetivo contrário ao nosso que visa a classificação de gênero em imagens sem restrições.

3.2. Classificação de gênero

Como forma de avaliação de performance da abordagem desenvolvida, utilizou-se dois conjuntos de dados adquiridos em ambientes sem restrições (*in the wild*): Large-scale CelebFaces Attributes (CelebA) e Faces of the World (FotW). Os experimentos foram conduzidos em duas fases distintas: a primeira avaliou a classificação de gênero utilizando apenas classificadores treinados com uma única parte facial separadamente, com a arquitetura proposta por Zavan *et al.* [Zavan et al.] e, a segunda fase realizou-se a classificação com o classificador treinado a partir da fusão das características das partes faciais (olhos + nariz + boca), utilizando a arquitetura apresentada na Figura 1. Durante o processo de treinamento, apenas duas classes foram utilizadas, uma indicando o gênero masculino (0) e a outra o feminino (1). Todos os resultados apresentados foram obtidos com a detecção automática da face e das partes faciais.

3.3. CelebA dataset

A Large-scale CelebFaces Attributes (CelebA) [Liu et al. 2015] é composta por imagens de celebridades, cada uma contendo 40 atributos anotados manualmente (por exemplo, óculos de sol, gênero). As imagens deste conjunto apresentam uma grande variação de pose e de fundo. A CelebA é formada por 202.599 imagens de faces obtidas através de 10.177 indivíduos diferentes e 5 anotações de *landmarks*. Este conjunto é dividido em três subconjuntos: treinamento (162.770 imagens), validação (19.867 imagens) e teste (19.962 imagens). Para os experimentos empregou-se a divisão original dos subconjuntos presentes na CelebA dataset. A Tabela 3 ilustra um comparativo entre a arquitetura proposta e o método proposto por [Zavan et al.].

Tabela 3. Taxa de acurácia de classificação de gênero

CelebA Dataset	Olhos	Nariz	Boca
NosePose [Zavan et al.]	92.45%	90.98%	91.65%
CNNNFusion-gender	98.13%	97.90%	97.40%

Neste comparativo, a CNNFusion-Gender apresentou-se superior em relação ao método proposto por [Zavan et al.]. Além disso, outro comparativo foi realizado em relação a esta abordagem com demais métodos considerados estado-da-arte presentes na literatura, que visam também a classificação de gênero. A Tabela 4 destaca a comparação da CNNFusion-Gender entre outros métodos.

Tabela 4. Classificação de Gênero - Comparativo com o métodos estado da arte

Métodos	Gênero
FaceTracker [Kumar et al. 2008]	91%
Panda-1 [Zhang et al. 2014]	97%
Panda-w [Zhang et al. 2014]	93%
Li and Zang [Li and Zhang 2013]+ANet [Liu et al. 2015]	95%
MT-RBM [Ehrlich et al. 2016]	90%
Walk & Learn [Wang et al. 2016]	96%
LNets+ANet [Liu et al. 2015]	98%
HyperFace [Ranjan et al. 2016a]	97%
Rajanet <i>al.</i> [Ranjan et al. 2016b]	99%
CNNFusion-Gender (Olhos)	98.13%
CNNFusion-Gender (Nariz)	97.90%
CNNFusion-Gender (Boca)	97.40%
CNNFusion-Gender (Regra de Votação [Kittler et al. 1998])	98.34%

Com base nos experimentos realizados na CelebA [Liu et al. 2015], comprovou-se também que a CNNFusion-Gender é eficiente para o processo de classificação de gênero utilizando apenas uma única parte facial por meio da combinação de características e, também obteve resultados satisfatórios em relação a maioria dos métodos de classificação.

Obteve-se resultados superiores em relação a maioria dos métodos, quando comparado com a classificação realizada através dos olhos (98.13%) e com a aplicação da regra de votação (98.34)%, exceto ao método proposto por Rajan *et al.* [Ranjan et al. 2016b] que apresenta 99% de acerto.

Ao analisar os resultados, os olhos demonstraram mais discriminantes para o processo de classificação, quando analisado em relação a CelebA dataset. Durante a etapa de treinamento, o olho esquerdo e direito foram treinados juntos, através da união da regiões dos mesmos, passando assim a região total como entrada na CNNFusion-Gender.

Além do comparativo em relação a classificação de gênero, analisamos a performance da CNNFusion-Gender na classificação de um atributo específico (gênero masculino) presente entre os 40 atributos anotados da CelebA dataset. A Tabela 5 exhibe o comparativo entre os métodos utilizados para classificação de gênero masculino na CelebA dataset.

Tabela 5. Comparativo de classificação do gênero masculino no conjunto de dados CelebA

Métodos	Gênero Masculino
FaceTracker [Kumar et al. 2008]	91%
Panda-1 [Zhang et al. 2014]	97%
Panda-w [Zhang et al. 2014]	93%
Li and Zang [Li and Zhang 2013]+ANet [Liu et al. 2015]	95%
LNets+ANet [Liu et al. 2015]	98%
CNNFusion-Gender (Olhos)	97.70%
CNNFusion-Gender (Nariz)	97.40%
CNNFusion-Gender (Boca)	96.50%
CNNFusion-Gender (Regra de Votação [Kittler et al. 1998])	98%

Ao comparar a abordagem proposta, obteve-se resultados inferiores apenas em relação ao método [Liu et al. 2015], quando comparados com olhos, boca e nariz. No entanto, ao comparar com o resultado gerado com a regra de votação, alcançamos o mesmo resultado apresentado por [Liu et al. 2015] (98%).

3.4. Faces of the World

A Faces of the World (FotW) [Escalera et al. 2016] consiste em um conjunto de imagens criado para a realização do *ChaLearn Looking at People and Faces of the World Challenge and Workshop*, utilizado para classificação de sorriso e gênero. Este conjunto de dados é composto por três subconjunto: Treinamento (6.171 imagens), Validação (3.086 imagens) e teste (8.505 imagens).

Neste conjunto de dados, realizou-se um comparativo com os demais métodos de classificação de gênero, que utilizam a face inteira para tal processo. Para o experimento, utilizou-se a divisão original do conjunto de imagens. A Tabela 6 exibe o comparativo da classificação de gênero com demais métodos estado-da-arte.

Tabela 6. Comparativo de classificação de gênero na base de dados FotW

Métodos	Gênero
MT-RBM [Ehrlich et al. 2016]	71.70%
CMP+ETH [Uřičář et al. 2016]	89.15%
DeepBE [Li et al. 2016]	90.44%
SIAT-MMLAB [Zhang et al. 2016]	91.66%
Rajan <i>et al.</i> [Ranjan et al. 2016b]	93.12%
CNNFusion-Gender (Olhos)	90.08%
CNNFusion-Gender (Nariz)	92.55%
CNNFusion-Gender (Boca)	91.52%
CNNFusion-Gender (Regra de Votação) [Kittler et al. 1998])	92.96%

Os resultados obtidos foram superiores aos métodos [Ehrlich et al. 2016, Uřičář et al. 2016, Li et al. 2016, Zhang et al. 2016] quando comparado com a CNNFusion-Gender (Regra de Votação), perdendo apenas para o método proposto por [Ranjan et al. 2016b]. Todavia, ao comparamos a CNNFusion-Gender(Olhos),

obteve-se resultado superior apenas no método [Ehrlich et al. 2016, Uřičář et al. 2016]. Este desempenho deve-se ao fato de várias imagens apresentarem algum tipo de acessório na região dos olhos, por exemplo, óculos, cabelo. Neste conjunto de dados, o nariz apresentou-se mais discriminante para o processo de classificação de gênero.

4. Conclusão

Neste trabalho, investigamos partes faciais, da mesma forma que técnicas para aprimorar a classificação de gênero. Sendo assim, a CNNFusion-Gender demonstrou-se competitiva em relação aos demais métodos estado da arte. Experimentos realizados em conjuntos de dados sem restrições CelebA e FotW comprovaram o alto desempenho da nova arquitetura em diferentes cenários. A abordagem apresentou resultados satisfatórios quando comparado aos demais métodos considerados estado-da-arte em classificação de gênero, evidenciando assim a eficácia do uso de apenas partes faciais, ao invés da face toda para o processo de classificação de gênero.

Por fim, o presente trabalho comprovou por meio dos experimentos que a taxa de acerto de classificação usando partes faciais é tão eficiente quanto utilizado a face inteira, sem que haja a necessidade do uso de métodos de alinhamento de face. Para trabalhos futuros visa a aplicação da CNNFusion-Gender em vídeos e imagens 3D.

Referências

- Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., and Kumar, N. (2013). Localizing parts of faces using a consensus of exemplars. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2930–2940.
- Ehrlich, M., Shields, T. J., Almaev, T., and Amer, M. R. (2016). Facial attributes classification using multi-task representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 47–55.
- Escalera, S., Torres Torres, M., Martinez, B., Baró, X., Jair Escalante, H., Guyon, I., Tzimiropoulos, G., Corneou, C., Oliu, M., Ali Bagheri, M., et al. (2016). Chalearn looking at people and faces of the world: Face analysis workshop and challenge 2016. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8.
- Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448.
- Goutte, C. and Gaussier, E. (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European Conference on Information Retrieval*, pages 345–359. Springer.
- Gu, L. and Kanade, T. (2008). A generative shape regularization model for robust face alignment. In *European Conference on Computer Vision*, pages 413–426. Springer.
- Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.
- Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K., Eggert, D. W., Fitzgibbon, A., and Fisher, R. B. (1996). An experimental comparison of range image segmentation algorithms. *IEEE transactions on pattern analysis and machine intelligence*, 18(7):673–689.

- Hu, G., Yang, Y., Yi, D., Kittler, J., Christmas, W., Li, S. Z., and Hospedales, T. (2015). When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 142–150.
- Kittler, J., Hatef, M., Duin, R. P., and Matas, J. (1998). On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence*, 20(3):226–239.
- Kumar, N., Belhumeur, P., and Nayar, S. (2008). Facetracer: A search engine for large collections of images with faces. In *European conference on computer vision*, pages 340–353. Springer.
- Kumar, N., Berg, A. C., Belhumeur, P. N., and Nayar, S. K. (2009). Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 365–372. IEEE.
- Le, V., Brandt, J., Lin, Z., Bourdev, L., and Huang, T. S. (2012). Interactive facial feature localization. In *European Conference on Computer Vision*, pages 679–692. Springer.
- Levi, G. and Hassner, T. (2015). Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 34–42.
- Li, B., Lian, X.-C., and Lu, B.-L. (2012). Gender classification by combining clothing, hair and facial component classifiers. *Neurocomputing*, 76(1):18–27.
- Li, C., Kang, Q., Ge, G., Song, Q., Lu, H., and Cheng, J. (2016). Deepbe: Learning deep binary encoding for multi-label classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 39–46.
- Li, J. and Zhang, Y. (2013). Learning surf cascade for fast and accurate object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3468–3475.
- Liao, Z., Petridis, S., and Pantic, M. (2017). Local deep neural networks for age and gender classification. *arXiv preprint arXiv:1703.08497*.
- Liu, C., Yuen, J., and Torralba, A. (2011). Nonparametric scene parsing via label transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2368–2382.
- Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738.
- Manyam, O. K., Kumar, N., Belhumeur, P., and Kriegman, D. (2011). Two faces are better than one: Face recognition in group photographs. In *Biometrics (IJCB), 2011 International Joint Conference on*, pages 1–8. IEEE.
- Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814.
- Ranjan, R., Patel, V. M., and Chellappa, R. (2016a). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249*.

- Ranjan, R., Sankaranarayanan, S., Castillo, C. D., and Chellappa, R. (2016b). An all-in-one convolutional neural network for face analysis. *arXiv preprint arXiv:1611.00851*.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.
- Saragih, J. M., Lucey, S., and Cohn, J. F. (2009). Face alignment through subspace constrained mean-shifts. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1034–1041. IEEE.
- Smith, B. M., Zhang, L., Brandt, J., Lin, Z., and Yang, J. (2013). Exemplar-based face parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3484–3491.
- Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.
- Song, F., Tan, X., and Chen, S. (2014). Exploiting relationship between attributes for improved face verification. *Computer Vision and Image Understanding*, 122:143–154.
- Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.
- Uříčář, M., Timofte, R., Rothe, R., Matas, J., and Van Gool, L. (2016). Structured output svm prediction of apparent age, gender and smile from deep features.
- Wang, J., Cheng, Y., and Feris, R. S. (2016). Walk and learn: Facial attribute representation learning from egocentric video and contextual data. *arXiv preprint arXiv:1604.06433*.
- Zavan, F. H., Nascimento, A. C., Bellon, O. R., and Silva, L. Nosepose: a competitive, landmark-free methodology for head pose estimation in the wild.
- Zhang, K., Tan, L., Li, Z., and Qiao, Y. (2016). Gender and smile classification using deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 34–38.
- Zhang, N., Paluri, M., Ranzato, M., Darrell, T., and Bourdev, L. (2014). Panda: Pose aligned networks for deep attribute modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1637–1644.
- Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE.

Segmentação das fases do gesto com máquinas de vetores suporte: o impacto da subjetividade e de diferentes rotulações

Jallysson M. Rocha, Ricardo A. Feitosa, Clodoaldo A. M. Lima, Sarajane M. Peres

¹Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
São Paulo – SP – Brasil

{jallysson, ricardoasfa, c.lima, sarajane}@usp.br

Resumo. *Com os avanços na automação a interpretação de gestos é esperado que o desenvolvimento de sistemas que suportem a comunicação entre pessoas e entre pessoas e máquinas também evolua. A gesticulação natural desempenha um papel fundamental neste contexto. Porém, a automação da interpretação da gesticulação natural envolve análises subjetivas que impactam a qualidade dos sistemas pretendidos na área. Este artigo apresenta um estudo que confirma a influência da subjetividade de rotulação nessa automação e indica o uso de classificadores como parâmetro de qualidade na análise da gesticulação. O estudo é desenvolvido no contexto da segmentação das fases do gesto, modelada sob um problema de classificação resolvido com Máquinas de Vetores Suporte.*

Abstract. *With the advances in the automated interpretation of gestures, the development of systems for communication between people and between people and machines are expected to evolving. Natural gesticulation plays a key role in such a context. However, the automation in the interpretation of natural gesticulation involves subjective analyzes that impact the quality of the systems in the area. This study confirms the influence of labeling subjectivity in such an automation task and suggest the use of classifiers as a quality parameter in the analysis of gesticulation. The study automates gesture phase segmentation through a classification problem solved with Support Vector Machines.*

1. Introdução

Atualmente, pesquisas na área de análise de gestos têm recebido especial atenção de pesquisadores na computação. Esforços de pesquisa têm focado no aprimoramento da interação humano-computador, e na criação de aplicações de diversas naturezas. Alguns exemplos de aplicação da análise de gestos são: manipulação de objetos digitais [Spano et al. 2012] ou equipamentos reais [Pickering 2005] usando gestos manuais ou corporais, interface multimodais para jogos [Liu and Kavakli 2010], suporte à comunicação via língua de sinais [Dong et al. 2015, Lee et al. 2016, Freitas et al. 2017], incorporação de gesticulação natural em robôs [Salem et al. 2012, Bremner et al. 2009].

A análise da gesticulação natural vem ganhando espaço entre os pesquisadores como uma sub-área de estudo em análise de gestos. A automação de tal análise representa um desafio porque essa classe de gestos é, segundo [Kim et al. 2007], desprovida de sentido específico e bastante dependente de diversidades culturais. Os estudos nessa área são geralmente orientados à análise do comportamento humano e permitem a formulação

de problemas sofisticados como a tomada de decisão sobre intencionalidade dos gestos [Jacob and Wachs 2014], a autenticação biométrica comportamental [Drosou et al. 2012] e a análise do discurso [Madeo et al. 2016]. Particularmente, análise dos gestos enquanto seus constituintes é estudada na Linguística, inserindo os gestos como parte do sistema linguístico. Nesse contexto, ferramentas computacionais de apoio à interpretação dos gestos [Allwood et al. 2007, Brugman and Russel 2004, Gibbon et al. 2003, Kipp 2012, Martell 2005, Maricchiolo et al. 2012] fornecem recursos para suporte de anotação manual dos gestos seguindo *frameworks* e gramáticas específicas.

Dentro do contexto da gesticulação natural, a segmentação das fases do gesto os analisa em termos de unidades gestuais – excursão realizada pelas mãos caracterizando um movimento que inicia e finaliza em uma posição de descanso das mãos. A unidade gestual é composta por fases gestuais [Kendon 1980, McNeill 1992] – segmentos do movimento que caracterizam períodos do gesto denominados *preparação*, *pré-stroke hold*, *stroke*, *pós-stroke hold* e *retração* [Kendon 1980, McNeill 1992]. Ainda, [Kita et al. 1998] também definem a fase de *hold independente* e uma gramática que semi-formaliza a ocorrência das fases dentro de uma unidade gestual.

Realizar a segmentação dos gestos em suas fases de forma manual é um processo que exige muito tempo e atenção de um especialista [Quek et al. 2002], principalmente por conta da alta subjetividade inerente à tarefa e à necessidade de manter um padrão de tomada de decisão durante horas de análise de um vídeo contendo a gesticulação. A automação da tarefa pode melhorar a padronização do resultado e facilitar o trabalho dos especialistas, que podem seguir com análises mais avançadas a partir dos resultados de segmentação fornecidos por um processo automatizado [Madeo 2013].

Uma maneira de implementar a segmentação automática das fases do gestos é sob um problema de classificação de *frames* de vídeos dentro das diferentes fases [Madeo et al. 2016, Ramakrishnan 2011]. Porém, classificadores são dependentes de conjuntos de dados rotulado, e a subjetividade de interpretação na identificação das fases dificulta a obtenção de rotulações consistentes, e conseqüentemente, dificulta a obtenção de classificadores eficazes para a tarefa em questão.

Este artigo tem por objetivo apresentar um estudo que aplica Máquinas de Vetores Suporte [Haykin 1999, Lima 2004] na tarefa de segmentação das fases do gestos e discute a influência da subjetividade de rotulação no desempenho dos classificadores obtidos. Para isso, o artigo é organizado em: apresentação de conceitos teóricos (Seção 2); definição do problema de classificação (Seção 3) e de critérios de avaliação (Seção 4); experimentação e análise de resultados (Seção 5); considerações finais (Seção 6).

2. Teoria dos Gestos

A área de aplicação deste artigo está relacionada à análise de comportamento humano. Do ponto de vista da Ciência da Computação, esta área deve ser desenvolvida, como discutido por [Pantic et al. 2007], como uma abordagem que se preocupa em modelar a comunicação (qual é o tipo de mensagem utilizada: mensagem linguística, sinal não linguístico, emoção, atitude), como a informação é passada (expressão facial, gesto, movimentos de cabeça) e por que a informação é passada (o contexto da ação: onde o usuário está, o que está fazendo, se há outras pessoas envolvidas). Na Linguística e na Psicolinguística, uma maneira singular de estudar a comunicação é pautada na

análise de padrões espaço-temporais gerados pela realização de “movimentos” e “gestos”¹ e pela inserção e análise da significação destas variáveis no contexto comportamental [McCleary and Viotti 2009]. Diferentes estudos que revisam o estado da arte em análise de gestos trazem as diferentes facetas e desafios da área. Como forma de acessar um espectro bastante abrangente sobre esta área, é interessante considerar algumas revisões que têm sido publicadas na área [Pickering 2005, Mitra and Acharya 2007, Sowa 2008, Moni and Ali 2009, Liu and Kavakli 2010, Madeo et al. 2013, Madeo et al. 2017].

O contexto do presente artigo é a área de análise de gestos sob a visão da Linguística e Psicolinguística que dá embasamento ao entendimento da produção e compreensão da linguagem [Garnham 1994]. No estudo da linguagem, para [McNeill 1992] e [Kendon 1980], é necessário considerar os gestos, uma vez que eles e fala são parte de um mesmo sistema linguístico, oriundos da mesma fonte semântica. Uma das linhas de análise de gestos se baseia no estudo da divisão deles em fases [Kendon 1980, Kita et al. 1998].

As fases do gesto estão diretamente ligadas ao que [Kendon 2005] chamou de “movement excursions”. Segundo o mesmo autor, uma pessoa faz uma ou várias “movement excursions” durante um discurso. Estas excursões se referem a movimentos das mãos de alguma “posição de descanso” para uma região no espaço onde um movimento “importante” ocorre, e depois, de volta para a posição de descanso. Toda a excursão (ou trajetória) do movimento é chamada de “unidade gestual”, enquanto as posições que as mãos assumem entre elas são chamadas de “posições de descanso”. Uma unidade gestual pode ser segmentada em fases do gesto, que são representadas por²:

- preparação – as mãos se movem na direção em que ocorrerá o gesto;
- *pré-stroke hold* – uma breve pausa no final da preparação mantendo a mesma configuração das mãos;
- *stroke* – ocorre obrigatoriamente no gesto e expressa o seu significado;
- *pós-stroke hold* – uma breve pausa ao final do *stroke* mantendo a mesma configuração das mãos;
- retração – momento em que as mãos retornam a posição de descanso;
- [Kita et al. 1998] também definem a fase de *hold*, que caracteriza-se por ser uma fase que não apresenta movimento, mas que expressa significado.

Na figura 1(a), uma amostragem de *frames* de vídeo ilustra posições de descanso (três primeiros e três últimos *frames*) e uma unidade gestual (do quarto ao décimo segundo *frame*, em intervalos de três, as fases de preparação, *stroke* e retração estão representadas). Na figura 1(b), a fase de *hold* é representada do terceiro ao sétimo *frame*.

3. Definição do problema

Neste artigo, a segmentação automática das fases do gesto é obtida por meio da análise de uma sequência s de *frames* $\{f_1 \cdots f_n\}$ consecutivos. O problema pode ser modelado

¹Movimentos (corporais), embora possam fazer parte de um gesto, não podem ser considerados gestos em si. Porém, na linha de pesquisa que está sendo seguida neste artigo, um gesto sempre envolve um movimento. Dessa forma, no presente texto, o conceito de gestos será usado de maneira a englobar o conceito de movimento e, daqui para frente, seguir-se-á falando em análise de gestos apenas.

²Embora sejam seis as fases do gesto, a análise deste estudo exclui as fases de pré e pós *stroke hold* devido à sua elevada complexidade de análise.

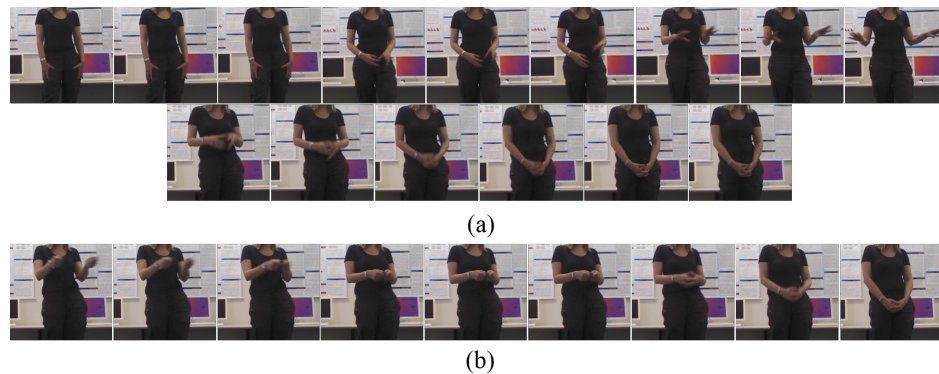


Figura 1. Fases do gesto: *Frames* amostrados de um vídeo contendo gesticulação natural referente à contação de histórias

como uma tarefa de classificação na qual, dada a sequência s como entrada, o classificador deve classificar um *frame* de interesse $f_{int} \in s$ em uma das classes $c_i \in \{D, P, H, S, R\}$, que correspondem respectivamente a períodos de descanso, e às fases preparação, *hold*, *stroke* e retração.

A manifestação de uma fase do gesto se dá via um comportamento da mão do gesticulador no tempo e no espaço. Sendo assim, para tratar o aspecto temporal, é preciso ou adotar uma técnica para construção do classificador que seja capaz de processar informação temporal, ou usar uma representação vetorial do tempo. Resultados apresentados na literatura mostram que a representação vetorial do tempo permite o alcance de resultados melhores, e portanto ela é adotada neste trabalho seguindo o exposto em [Madedo 2013]. Essa representação consiste em sub-sequências s' que são obtidas da sequência s por meio de janelas deslizantes, iniciadas, centradas ou finalizadas em f_{int} . Dessa forma, desconsiderando um pequeno trecho inicial e/ou final do vídeo, todos os *frames* são apresentados ao classificador, acompanhados de *frames* anteriores e/ou posteriores, representando informação passada e futura.

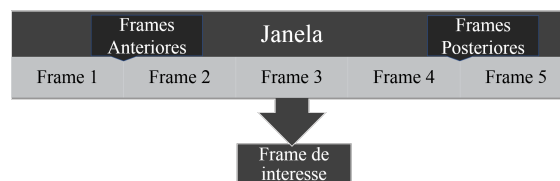


Figura 2. Uma janela de cinco *frames* e *frame* de interesse no meio

Ainda, é preciso dividir o problema em subtarefas de classificação binária para que ele se adeque à técnica de Máquina de Vetores Suporte. Diferentes estratégias podem ser usadas para alcançar esse objetivo. Neste trabalho, a estratégia “um-contra-todos” foi aplicada, de forma que o problema de classificação multi-classe é dividido em cinco problemas de classificação binária.

Para suportar o aprendizado supervisionado de um classificador de fases do gestos, é necessário considerar um vídeo contendo contação de histórias rotulado de acordo com as fases. Esse é o ponto em que se pode explorar o problema da subjetividade de análise.

Para este estudo, um vídeo foi rotulado por dois rotuladores, e mais três rotulações foram obtidas a partir da combinação das rotulações manuais.

4. Critérios de avaliação

A fim de analisar o impacto da subjetividade e de diferentes rotulações sobre o desempenho do classificador, alguns critérios de avaliação foram aplicadas: análise de concordância de rotulação, análise quantitativa do desempenho de classificação por *frame* apresentado pelos classificadores e análise qualitativa das segmentações produzidas.

A análise de concordância de rotulação foi realizada a partir do coeficiente *Alpha Krippendorff* [Krippendorff 2007]. Esse coeficiente permite medir a confiabilidade dos rotuladores, e pode ser obtido por $\alpha = 1 - \frac{D_o}{D_e}$, em que D_o é a quantidade de desacordos observados, e D_e é a quantidade de acordos esperados. Em detalhes: $D_o : \frac{1}{n} \sum_c \sum_k O_{ck} \delta_{ck}^2$ e $D_e : \frac{1}{n(n-1)} \sum_c \sum_k n_c * n_k \delta_{ck}^2$, em que O_{ck} , n_c , n_k e n são as frequências de valores em uma matriz de coincidências; δ é uma função que implementa uma métrica adequada para diferentes tipos de dados [Krippendorff 2007].

O resultado do coeficiente varia entre -1 e 1 , considerando valores negativos como evidências de dados insuficientes para análise ou concordância aleatória entre os codificadores. Os valores positivos para esse coeficiente são interpretados da seguinte forma: entre 0 e $0,2$ indica concordância fraca; entre $0,2$ e $0,4$ razoável; entre $0,4$ e $0,6$ moderada; entre $0,6$ e $0,8$ substancial e entre $0,8$ e 1 a concordância pode ser considerada perfeita. A escolha desse coeficiente se deu por ser bastante utilizado na literatura de análises de concordância e por ser capaz de minimizar o viés dos codificadores [Artstein and Poesio 2008].

A análise quantitativa do desempenho dos classificadores foi realizada a partir das medidas precisão, revocação e *F-score* [Fawcett 2006], derivadas das matrizes de confusão binárias referente aos modelos binários criados sob a estratégia “um-contra-todos”. Diante de tais medidas, o desempenho de classificação dos *frames* em cada uma das fases do gestos foi avaliado, e os conjuntos de classificadores com melhores desempenhos nessas medidas foram selecionados para passar por análises mais detalhadas (conforme descrito na seção 5).

Para a avaliação final dos modelos, considerando a resolução do problema de classificação multi-classe, foi necessário combinar os resultados obtidos em cada sub-problema de classificação binária. Para isso, foram comparadas as respostas oferecidas pelo cinco classificadores para cada *frame* de teste, considerando a distância dos dados de teste para os hiperplanos de separação. A resposta final dos classificadores multi-classe foram usadas para fornecer o erro final (**EF** – taxa de *frames* classificados incorretamente) do modelo e para avaliação das segmentações produzidas.

A análise de qualidade de segmentação representa uma avaliação de qualidade mais próxima da visão do especialista (e.g. um linguísta). Para quantificar o desempenho dos classificadores sob esse ponto de vista, os seguintes erros foram computados:

- *Erros de transição (ET)*, ou seja, erros de classificação cometidos na transição (borda) entre fases. Esses erros são similares à discordância de rotulação entre dois ou mais rotuladores, e possuem uma criticidade menor do que os erros interno;

- **Erros internos (EI)**, ou seja, erros de classificação cometidos dentro de uma fase do gesto;
- **Erros de inserção (INS)**, os quais ocorrem quando o classificador insere uma fase do gesto fora da unidade gestual (dentro de um período de descanso);
- **Erros de exclusão (EXC)**, os quais ocorrem quando uma fase do gesto não é reconhecida pelo classificador (um período de descanso é inserido dentro de uma unidade gestual);
- **Erros de substituição (SUB)**, os quais ocorrem quando a resposta do classificador substitui uma fase do gesto por outra fase do gesto.

A quantificação dos erros ET requerem o estabelecimento do tamanho da borda de uma fase. Supondo que a borda da fase seja dimensionada para X frames, então os erros de transição podem ocorrer em um intervalo de $2 * X$ frames consecutivos entre duas fases do gesto (X frames em uma fase e X frames em outra fase). Todos os outros erros são considerados erros EI. A quantificação dos erros INS, EXC, e SUB requerem o encontro de Y frames de uma mesma fase dentro de outra, conforme a definição de cada erro. Neste trabalho $X = 5$ e $Y = 80\%$ de frames na fase.

5. Experimentos e resultados

Visto que máquinas de vetores suporte implementam aprendizado indutivo supervisionado, a criação de modelos classificadores sob ela exigem um conjunto de exemplos rotulado. Para estes experimentos, foi usado como instância de treino e teste, um vídeo de contação de história de 2.783 frames de extensão. O vídeo foi capturado com o uso do sensor *Microsoft Kinect for Windows v1* e cada frame foi manualmente rotulado em uma fase do gesto por dois rotuladores (**R1** e **R2**), seguindo a gramática de [Kita et al. 1998]³. A rotulação é equivalente à segmentação das fases.

Para propiciar a análise do impacto da subjetividade nos resultados de classificação, foram criadas três novas rotulações. Tais rotulações (**R3**, **R4** e **R5**) tiveram o objetivo de reduzir a discordância entre os rotuladores e, conseqüentemente, reduzir o nível de subjetividade inerente à segmentação das fases do gesto. As novas rotulações são decorrentes de decisões tomadas em conjunto pelos dois rotuladores, considerando: (a) a discordância na rotulação de frames referentes a períodos de descanso; (b) a discordância na rotulação de frames referentes à fase *stroke*.

No caso (a), a subjetividade gera discordâncias de rotulação que resultam na inserção de fases do gesto dentro do períodos de descanso. Para minimizar esse problema, dois padrões de correção foram seguidos, em comum acordo com os especialistas a partir da análise do vídeo:

- R3** há uma mudança na posição de descanso erroneamente interpretada como um gesto. Nesse caso é realizada a substituição dos rótulos de fases do gestos por rótulos de período de descanso.
- R5** o gesticulador inicia um gesto, mas recua como se desistisse de realiza-lo. Nesse caso, os respectivos frames são rotulados como fases de preparação e retração.

³O vídeo usado nesse experimentos é parte de um corpus de gesticulação natural que está sendo construído sob um projeto aprovado por um comitê de ética em pesquisa. O projeto está registrado na Plataforma Brasil sob Certificado de Apresentação para Apreciação Ética (CAAE) 53366516.9.0000.5390.

No caso (b), ocorrem situações semelhantes às anteriores, contudo dentro da fase *stroke*. Frequentemente, as discordâncias ocorrem porque o gesticulador transita de um gesto significativo para outro, possibilitando duas interpretações – um único (e longo) *stroke*, dois *strokes* separados por uma fase de preparação (ou retração) ou dois *strokes* separados com fases de preparação e retração. Para minimizar esse problema, três padrões de correção foram seguidos, em acordo com os especialistas a partir da análise do vídeo:

- R3** é realizada a substituição dos rótulos das fases preparação e retração por rótulos da fase *stroke*.
- R4** acata-se a fase de preparação como a fase de transição entre os dois *strokes* e o rótulo preparação é atribuído aos *frames* considerados como transição entre os *strokes*; mudança de posição de descanso são consideradas como em **R3**.
- R5** acata-se as fases de retração e preparação como fases de transição entre os dois *strokes* e os rótulos referentes a essas fases são adequadamente atribuídos aos *frames* considerados como transição entre os *strokes*.

5.1. Criação e seleção de classificadores

A primeira parte do experimento de construção de classificadores consistiu em explorar o espaço de representação do gesto e o espaço de busca de parâmetros da máquina de vetores suporte. A exploração desses espaços se deu por meio de uma busca em grade considerando os valores listados nas tabelas 1 e 2. Devido ao tamanho dos espaços de buscas, apenas a rotulação **R1** foi considerada nessa parte do estudo. Essa rotulação foi a escolhida por ter sido feita pelo rotulador mais experiente. Tanto na geração desses classificadores quanto dos próximos, na segunda parte do experimento, os 70% *frames* iniciais do vídeo foram usados no treinamento dos modelos e os 30% *frames* finais foram usados no teste.

Tabela 1. Parâmetros que formam o espaço de representação para os gestos

	Espaço de representação
Pontos de articulação	mãos, pulsos e cotovelos direitos e esquerdos
Filtro (para suavização do sinal)	Wavelet com função <i>db2</i> e nível 2
Características	coordenadas <i>x,y,z</i> , velocidade e aceleração escalar e vetorial
Número de <i>frames</i> na janela	10, 20, 30, 40, 50, 60, 70, 80, 90 e 100
Posição de f_{int} na janela	início, meio e fim

Tabela 2. Parâmetros que formam o espaço de representação para os gestos

	Espaço de parâmetros
Função <i>kernel</i>	Função de base radial
Parâmetro do <i>kernel</i>	2^{-5} a 2^5 , variando o expoente em um passo de tamanho 1
Margem	1, 100, 500, 1000

Os classificadores foram analisados em termos das medidas de revocação, precisão e F-score. A escolha dos melhores classificadores foi realizada com base na capacidade de alcançar desempenho acima de 0,7 para o maior número de fases. Nessa análise, classificadores especializados nas fases de descanso, *stroke* e retração foram selecionados. Não foram obtidos classificadores com bom desempenho nas cinco fases.

Os melhores resultados foram obtidos sob representações de gestos que usam: mão direita, mão e pulso direitos; sinal sem suavização, sinal com suavização; coordenadas espaciais ou velocidade vetorial; janelas de 20, 30 e 40 *frames*; *frame* de interesse na posição central da janela. Em relação aos parâmetros do SVM, foi possível verificar que valores diferentes para o parâmetro do kernel são adequados para cada tipo de representação dos gestos: 2^{-5} e 2^{-4} para coordenadas espaciais, com ou sem suavização do sinal; 2^2 a 2^4 para velocidade vetorial sem suavização do sinal; e 2^{-1} a 2^4 para velocidade vetorial com suavização do sinal.

5.2. Criação de classificadores para diferentes rotulações do conjunto de dados

Dentro de espaços de busca de representação e de parâmetros do classificador menores, foi possível realizar um estudo sobre o impacto de diferentes rotulações sobre o desempenho de classificação. Novos classificadores foram gerados, sob o conjunto de parametrizações reduzido, usando o mesmo conjunto de treinamento e teste, porém considerando as rotulações **R2**, **R3**, **R4** e **R5**.

Em geral, nessa nova etapa de construção de classificadores, os melhores resultados foram obtidos sob a representação espacial dos gestos (coordenadas x , y , z), sinal suavizado e janela de 40 *frames*. Os desempenhos de classificação são listados na tabela 3, em termos do melhor e pior F-score obtidos.

Tabela 3. Desempenho dos classificadores para cada fase do gesto, em termos de F-score máximo e mínimo.

Rotulador	Descanso		Preparação		<i>Hold</i>		<i>Stroke</i>		Retração	
	max	min	max	min	max	min	max	min	max	min
R1	0,79	0,77	0,17	0,15	0,00	0,00	0,75	0,71	0,74	0,70
R2	0,78	0,71	0,43	0,26	0,38	0,24	0,60	0,54	0,49	0,27
R3	0,78	0,76	0,17	0,14	0,00	0,00	0,78	0,71	0,73	0,52
R4	0,79	0,76	0,25	0,19	0,03	0,00	0,69	0,60	0,69	0,53
R5	0,79	0,77	0,22	0,17	0,03	0,00	0,70	0,61	0,51	0,38

Tais resultados mostram que, em geral, não houve uma sensibilidade alta às adequações de rotulação realizadas. Os resultados obtidos com a rotulação **R1** são semelhantes àqueles obtidos com as rotulações **R3**, **R4** e **R5**. Porém, há algumas variações no resultado que nos permitem levantar algumas hipóteses:

- As rotulações **R1** e **R3** tendem a simplificar o problema, aproximando o problema de segmentação das fases do gesto ao problema de encontrar uma unidade gestual. O desbalanceamento de classes no conjunto de treinamento é acentuado.
- As rotulações **R4** e **R5** admitem maior variabilidade de fases em uma unidade gestual, e contribuem para o aumento de *frames* de preparação e retração. Sob essa rotulação os classificadores aprendem um pouco mais sobre a fase de preparação, porém há indícios que isso ocorre em detrimento do aprendizado para *stroke* e retração. No entanto, principalmente sob a rotulação **R5**, a perda de desempenho para retração é acentuada e não parece refletir em aprendizado para preparação.
- Finalmente, ao observar os resultados sob a rotulação **R2**, percebe-se que o desempenho do classificador é baixo para todas as fases da unidade gestual, contudo, há uma distribuição mais uniforme do aprendizado entre as classes.

A análise dos erros apresentados na seção 4 (EF, ET, EI, EXC, INS e SUB) foi realizada para os oito melhores classificadores obtidos. Os resultados são listados na tabela 4. Os erros EF, ET e EI são calculados em termos de porcentagem de *frames* classificados erroneamente no vídeo completo. Os erros INS, EXC e SUB são calculados em termos de porcentagem de *frames* envolvidos nesses erros.

Tabela 4. Análise qualitativa dos melhores modelos obtidos sob cada rotulação.

Rotulação	EF	ET	EI	EXC	INS	SUB
R1	35,15	1,55	33,61	4,46	0,00	9,38
R2	42,53	11,48	31,06	1,46	0,00	10,84
R3	34,88	3,19	31,69	5,56	0,00	8,47
R4	39,34	4,55	34,79	5,56	0,00	7,65
R5	39,34	4,74	34,61	5,56	0,00	7,65

Dado que os dados são desbalanceados (há uma quantidade maior de *frames* de descanso e de *stroke* do que de outras fases), os resultados da tabela 4 devem ser considerados apenas para fins de comparação do impacto das diferentes rotulações.

Em termos de erros que ocorrem nas transições entre fases, há um destaque negativo para o caso da rotulação **R2**. Embora o erro de transição seja aceitável e possa ser desconsiderado na análise do classificador, a porcentagem alta de erros desse tipo indica que o classificador não conseguiu capturar o padrão de rotulação do rotulador, sugerindo que se trata de um rotulador com mais subjetividade na tomada de decisão. No classificador criado com a rotulação **R1**, verifica-se que o padrão de rotulação foi mais facilmente aprendido. Há portanto um indicativo que o classificador tem, de fato, alta sensibilidade à rotulação nos *frames* de transição. Ainda, levanta-se a hipótese de que o classificador pode atuar também como um meio de analisar a qualidade do trabalho do rotulador.

Os erros EXC, INS e SUB se mantiveram parecidos para todas as rotulações, o que não permitiu levantar hipóteses sobre o impacto da subjetividade das rotulações. Contudo, nota-se que erros de substituição nas fases de preparação e retração diminuíram para o caso da rotulação **R5** (Tabela 5). Essa rotulação pode favorecer o aprendizado dessas fases por torná-la mais representativa no conjunto de treinamento, mas também pode estar contribuindo para uma melhor definição do padrão quando impõe que essas fases sejam representadas entre mudanças de posições de descanso e entre *strokes* consecutivos.

Tabela 5. Detalhamento dos erros de substituição por fase do gesto

Rotulação	Descanso	Preparação	Hold	Stroke	Retração
R1	0,00	36,89	63,11	0,00	0,00
R2	0,00	22,69	47,90	5,88	23,53
R3	0,00	30,11	69,89	0,00	0,00
R4	0,00	22,62	77,38	0,00	0,00
R5	0,00	10,71	77,38	0,00	11,90

6. Considerações finais

Este artigo relatou um estudo localizado na área de reconhecimento de padrões e segmentação automática das fases do gesto. Mais especificamente, foi objetivo deste

estudo, analisar o impacto de escolhas subjetivas, inerentes à análise de gestos, na construção de classificadores. As análises tomaram como base um extenso trabalho de calibração de parâmetros referentes à representação dos gestos e também referentes à máquina de vetores suporte. Então, assumindo um espaço menor de geração de classificadores, rotulações diferentes para um mesmo contexto de gesticulação natural foram avaliadas sob aspectos quantitativos e qualitativos.

Um estudo preliminar a este [Wagner et al. 2014] tratou um problema similar porém simplificado (tratando unidades gestuais), de forma que não é factível a realização de comparações diretas de desempenho e conclusões. Até onde foi possível verificar na literatura da área, este é o primeiro estudo que propôs uma análise sistemática e automatizada do problema da subjetividade de análise nas fases do gesto. Trata-se ainda de um estudo preliminar que precisa ser aprofundado e realizado em diferentes contextos de gesticulação natural. Contudo, o estudo já permitiu levantar algumas hipóteses.

Os experimentos e análises realizados confirmaram que o contexto da gesticulação natural representa um desafio para o reconhecimento automático de padrões. Os classificadores mostraram desempenhos diferenciados sob alguns aspectos quando consideradas as diferentes rotulações. O impacto da subjetividade de rotulação é evidente nas transições entre as fases, que é, segundo os rotuladores, o ponto de maior dificuldade de rotulação manual. Levantou-se a hipótese que o classificador possa ser usado como um parâmetro de qualidade de rotulação. Para além da questão da subjetividade, as diferentes rotulações permitiram também fundamentar a indicação de uso das fases de preparação e retração na demarcação de mudanças de posição de descanso e de *strokes* consecutivos.

Esses experimentos estão sendo estendidos para outros contextos de discurso como entrevistas e aulas, e também para outros tipos de representação de dados. Os próximos passos planejados para a continuidade deste estudo envolvem: incorporar testes estatísticos que permitam trazer mais informação para as análises realizadas; repetir esse tipo de análise usando outros classificadores e outros paradigmas de reconhecimento de padrões, como Cadeias de Markov ou reconhecedores de padrão sintáticos.

Referências

- Allwood, J., Cerrato, L., Jokinen, K., Navarretta, C., and Paggio, P. (2007). The mummin coding scheme for the annotation of feedback, turn management and sequencing phenomena. *Language Resources and Evaluation*, 41(3):273–287.
- Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.
- Bremner, P., Pipe, A. G., Frase, M., Subramanian, S., and Melhuish, C. (2009). Beat gesture generation rules for human-robot interaction. In *Proc. of the 18th IEEE Int. Symp. on Robot and Human Interactive Communication*, pages 1029–1034.
- Brugman, H. and Russel, A. (2004). Annotating multimedia/multi-modal resources with elan. In *Proc. of the 4th Int. Conf. on Language Resources and Evaluation*, pages 2065–2068.
- Dong, C., Leu, M. C., and Yin, Z. (2015). American sign language alphabet recognition using microsoft kinect. pages 44–52.

- Drosou, A., Ioannidis, D., Moustakas, K., and Tzovaras, D. (2012). Spatiotemporal analysis of human activities for biometric authentication. *Computer Vision and Image Understanding*, 116:411–421.
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874.
- Freitas, F. A., Peres, S. M., Lima, C. A. M., and Barbosa, F. V. (2017). Grammatical facial expression recognition in sign language discourse: a study at the syntax level. *Information Systems Frontiers*, pages 201–217. Online First.
- Garnham, A. (1994). *Psycholinguistics: Central Topics*. Routledge.
- Gibbon, D., Gut, U., Hell, B., Looks, K., and Thies, R. (2003). A computational model of arm gestures in conversation. In *Proc. of Eurospeech*, pages 813–816.
- Haykin, S. (1999). Multilayer perceptrons. *Neural networks: a comprehensive foundation*, 2:156–255.
- Jacob, M. G. and Wachs, J. P. (2014). Context-based hand gesture recognition for the operating room. *Pattern Recognition Letters*, 36:196–203.
- Kendon, A. (1980). Gesticulation and speech: Two aspects of the process of utterance. *The relationship of verbal and nonverbal communication*, 25:207–227.
- Kendon, A. (2005). *Gesture: Visible action as utterance*. Cambridge University Press.
- Kim, D., Song, J., and Kim, D. (2007). Simultaneous gesture segmentation and recognition based on forward spotting accumulative HMMs. *Pattern Recognition*, 40(11):3012–3026.
- Kipp, M. (2012). *Multimedia Annotation, Querying, and Analysis in Anvil*, chapter 19, pages 351–367. John Wiley e Sons, Inc.
- Kita, S., Van Gijn, I., and Van der Hulst, H. (1998). Movement phases in signs and co-speech gestures, and their transcription by human coders. In *Int. Gesture Workshop*, pages 23–35. Springer.
- Krippendorff, K. (2007). Computing krippendorff’s alpha reliability. *Departmental papers (ASC)*.
- Lee, G., h. Yeh, F., and Hslao, Y.-H. (2016). Kinect-based taiwanese sign-language recognition system. *Multimedia Tools and Applications*, 75:261–279.
- Lima, C. A. M. L. (2004). *Comitê de Máquinas: Uma Abordagem Unificada Empregando Máquinas de Vetores-Suporte*. PhD thesis, Universidade de São Paulo.
- Liu, J. and Kavakli, M. (2010). A survey of speech-hand gesture recognition for the development of multimodal interfaces in computer games. In *Multimedia and Expo (ICME)*, pages 1564–1569. IEEE International Conference on.
- Madeo, R. C., Lima, C. A., and Peres, S. M. (2017). Studies in automated hand gesture analysis: An overview of functional types and gesture phases. *Lang. Resour. Eval.*, 51(2):547–579.
- Madeo, R. C. B. (2013). Máquinas de vetores suporte e a análise de gestos: incorporando aspectos temporais. Master’s thesis, Universidade de São Paulo.

- Madeo, R. C. B., Peres, S. M., and de Moraes Lima, C. A. (2016). Gesture phase segmentation using support vector machines. *Expert Systems with Applications*, 56:100–115.
- Madeo, R. C. B., Wagner, P. K., and Peres, S. M. (2013). A review of temporal aspects of hand gesture analysis applied to discourse analysis and natural conversation. *Int. J. of Computer Science and Information Technology*, 5(4):1–20.
- Maricchiolo, F., Gnisci, A., and Bonaiuto, M. (2012). *Coding Hand Gestures: A Reliable Taxonomy and a Multi-media Support*, pages 405–416. Springer Berlin Heidelberg.
- Martell, C. H. (2005). *An Extensible, Kinematically-Based Gesture Annotation Scheme*, chapter Form, pages 79–95. Springer Netherlands.
- McCleary, L. and Viotti, E. (2009). Sign-gesture symbiosis in brazilian sign language narrative. *Meaning, form, and body*, pages 181–201.
- McNeill, D. (1992). *Hand and mind: What gestures reveal about thought*. University of Chicago press.
- Mitra, S. and Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3):311–324.
- Moni, M. and Ali, A. S. (2009). Hmm based hand gesture recognition: A review on techniques and approaches. In *IEEE Int. Conf. on Computer Science and Information Technology*, pages 433–437. IEEE.
- Pantic, M., Pentland, A., Nijholt, A., and Huang, T. S. (2007). Human computing and machine understanding of human behavior: a survey. In *Artificial Intelligence for Human Computing*, pages 47–71. Springer.
- Pickering, C. A. (2005). The search for a safer driver interface: a review of gesture recognition human machine interface. *Computing Control Engineering Journal*, 16(1):34–40.
- Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X.-F., Kirbas, C., McCullough, K. E., and Ansari, R. (2002). Multimodal human discourse: gesture and speech. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 9(3):171–193.
- Ramakrishnan, A. S. (2011). Segmentation of hand gestures using motion capture data. Master’s thesis, University of California, Davis.
- Salem, M., S. Kopp, I. W., Rohlfing, K., and Joublin, F. (2012). Generation and evaluation of communicative robot gesture. *Int. J. of Social Robotics*, 4(2):201–217.
- Sowa, T. (2008). The recognition and comprehension of hand gestures-a review and research agenda. In *Modeling Communication with Robots and Virtual Humans*, pages 38–56. Springer.
- Spano, L., Cisternino, A., and Paterno, F. (2012). A compositional model for gesture definition. In *Proc. of Int. Conf. on Human-Centered Software Engineering*, pages 34–52.
- Wagner, P. K., Peres, S. M., Madeo, R. C. B., Lima, C. A. M., and Freitas, F. A. (2014). Gesture unit segmentation using spatial-temporal information and machine learning. In *Florida Artificial Intelligence Research Society Conf.*, pages 101–106.

Concept Lattice Reduction using Fuzzy Data Clustering

Suzane Carol de Lima, Heloisa de Arruda Camargo

¹ Department of Computer Science - Federal University of São Carlos
Rodovia Washington Luís, km 235 - São Carlos - SP - Brazil

Email: lima.suzanecarol@gmail.com/heloisad@dc.ufscar.br

Abstract. *Formal concept analysis is considered an important formalism for knowledge representation, extraction and analysis which can be applied to many areas. The main challenge for the applications using formal concept analysis is that the structure of the concept lattice can become prohibitive large even for a small set of data due to the number of formal concepts identified. The objective of this paper is to propose and study a procedure to reduce the formal context by using fuzzy clustering algorithms, so that a reduced concept lattice can be generated from the reduced formal context. The experiments were run with two different fuzzy clustering algorithms, Fuzzy C Means and Possibilistic Fuzzy C Means, applied to data sets from the documents domain. Results show that the method proposed allows a considerable reduction on the amount of concepts.*

1. Introduction

The topic of knowledge representation is a fundamental concern for all types of knowledge-based systems. The Formal Concept Analysis (FCA) [Ganter and Wille 1999] is considered an important formalism for the representation of knowledge for it provides means to not only represent, but also to extract and analyze knowledge. FCA has been applied in several fields of research, such as data analysis, machine learning and software engineering, focusing on a variety of problems like information retrieval [Qadi et al. 2010], information visualization [Kumar and Srinivas 2010] and the automatic construction of ontologies [Maio et al. 2009].

FCA's main idea consists of transforming a given dataset, initially in the form of a relational binary matrix, called formal context, into a concept lattice, which is a complete lattice composed of the concepts extracted from the formal context. A formal concept is an ordered pair (A, B) where A is a set of objects and B is a set of attributes such that each object in A has all the attributes in B and each attribute in B is a common attribute to all the objects in A . The problem with using FCA in applications is that the number of concepts and relations can be prohibitively large, even for small datasets. All relations among concepts are represented in the concept lattice, making the computation cost very high.

Due to the relevance and potential applicability of the FCA formalism, the reduction of formal contexts and, consequently, of the generated concept lattice is a research topic that attracts the attention of the scientific community. Several techniques for concept lattice reduction have been proposed. According to [Dias and Vieira 2015], the techniques can be classified in three different classes: redundant information removal, simplification and selection. The techniques in the redundant information removal group seek to remove objects, attributes of relations that are redundant, that is, after their removal, the concept

lattice obtained is isomorphic to the original one [Medina 2012], [Li et al. 2011], [Wang and Zhang 2010]. Simplification techniques are the ones that construct an abstraction of the concept lattice eliminating small differences between concepts, attributes or objects, maintaining only the essential knowledge [Cheung and Vogel 2005], [Snasel et al. 2007], [Gajdoš et al. 2004], [Codocedo et al. 2011], [Kumar and Srinivas 2010]. The techniques in the selection group consist of selecting objects, attributes or concepts that are most relevant to the application. One of the problems of using this technique is the computational cost, since it is necessary to generate all the possible concepts [Belohlavek and Macko 2011], [Zhang et al. 2012].

Among the aforementioned methods, the ones of interest for the present work are those in the simplification category, especially the ones based on clustering techniques. These techniques can be applied to sets of objects, attributes or formal concepts in a very natural and flexible way. One of the earliest approaches consists of reducing the formal context with the technique of Singular Value Decomposition (SVD) [Cheung and Vogel 2005]. According to Kumar and Srinivas (2010), methods based on SVD have a high computation cost, making a method unviable mainly for large datasets. As an alternative, they used Fuzzy C Means algorithm to cluster objects [Kumar and Srinivas 2010]. However, the proposal was described considering two different numbers of clusters defined empirically, and a very small dataset for illustrative purposes. Considering the great variety of clustering algorithms available, either hard or fuzzy, the reduction of the formal context can be explored in several ways.

The goal of this paper is to propose a method to reduce the formal context and also the formal concepts generated from it by means of fuzzy clustering algorithms including a previous process of assessment of the best number of clusters. The experiments were run with two different fuzzy clustering algorithms, Fuzzy C Means (FCM) [Bezdek 1981] and Possibilistic Fuzzy C Means (PFCM) [Pal et al. 2005], applied to data sets from the documents domain.

The remainder of this work is organized as follows. Section 2 presents a review of the basic concepts related to FCA, fuzzy clustering algorithms and the evaluation index fuzzy silhouette. In section 3, the main proposals focusing on the reduction of concept lattices through simplification are described. The method proposed is presented in section 4. The experiments and results are described in section 5 and the conclusions are briefly discussed in section 6.

2. Basic Concepts

2.1. Formal Concept Analysis

FCA is a mathematical theory of data analysis using formal contexts, formal concepts and concept lattices. FCA analyzes data which describe relationship between a particular set of objects and a particular set of attributes. Concepts are sets of objects that share the same set of attributes. Concepts are related by a partial order relation that form a concept lattice and defines subconcepts-superconcepts relationships between concepts. Initially, we introduce the notion of formal context.

Definition 1. A formal context consists of a triple $C = (G, M, I)$, where G is a set of objects, M is the set of attributes and $I \subseteq G \times M$ is an incidence relation between G and M .

Table 1 presents an illustrative example of a formal context, in which the set of objects is $G = \{Dog, Cat, Monkey, Bird, Whale, Habbit, Shark, Elephant\}$ and the set of attributes is $M = \{fourLegs, fur, swin, fly, mammal, wings\}$. Each object is associated to a row and each attribute to a column. The entries of the table are marked with a cross when the object in the corresponding row has the attribute in the corresponding column.

Table 1. Formal Context on animal characteristics

	fourLegs	fur	swin	fly	mammal	wings
Dog	X	X			X	
Cat	X	X			X	
Monkey	X	X			X	
Bird				X		X
Whale			X		X	
Habbit	X	X			X	
Shark			X			
Elephant	X				X	

Every formal context induces a pair of operators, called derivation operators.

Definition 2. For a subset $A \subseteq G$ of objects, a subset $B \subseteq M$ of attributes the derivation operators \uparrow and \downarrow are defined as:

$$A^\uparrow = \{m \in M | gIm \text{ for all } g \in A\}$$

$$B^\downarrow = \{g \in G | gIm \text{ for all } m \in B\}$$

The operator \uparrow assigns subsets of M to subsets of G . A^\uparrow is the set of all attributes shared by all objects from A . The operator \downarrow assigns subsets of G to subsets of M . In other words, B^\downarrow is the set of all objects sharing all attributes from B . Some examples of applications of the derivation operators on the formal context presented in Table 1 are:

$$\{Dog, Cat\}^\uparrow = \{fourLegs, fur, mammal\}$$

$$\{Shark, Whale\}^\uparrow = \{swin\}$$

$$\{Dog, Cat, Monkey, Habbit\}^\uparrow = \{fourLegs, fur, mammal\}$$

$$\{mammal, swin\}^\downarrow = \{Whale\}$$

$$\{mammal\}^\downarrow = \{Dog, Cat, Monkey, Whale, Habbit, Elephant\}$$

$$\{fourLegs, fur, mammal\}^\downarrow = \{Dog, Cat, Monkey, Habbit\}$$

From a formal context and the derivation operators it is possible to extract formal concepts.

Definition 3. A formal concept in (G, M, I) is defined by the pair (A, B) if and only if $A \subseteq G$, $B \subseteq M$, and $A^\uparrow = B$ e $B^\downarrow = A$. A is called concept extension and B is

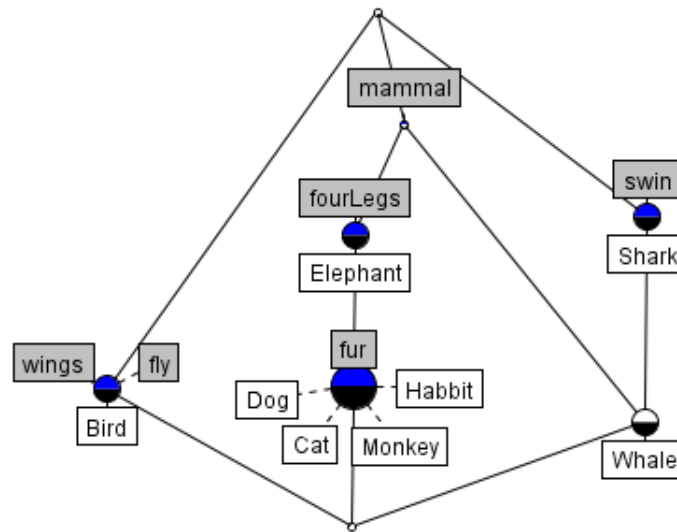


Figure 1. Concept Lattice about animals

called concept intention. Applying the operators in the formal context, one of the formal concepts found is:

$$\begin{aligned} \{Dog, Cat, Monkey, Habbit\}^\uparrow &= \{fourLegs, fur, mammal\} \\ \{fourLegs, fur, mammal\}^\downarrow &= \{Dog, Cat, Monkey, Habbit\} \\ (A, B) &= (\{Dog, Cat, Monkey, Habbit\}, \{fourLegs, fur, mammal\}) \end{aligned}$$

The set of formal concepts are organized in a structure called concept lattice.

Definition 4. A concept lattice is a taxonomic structure defined through the partial subconcept-superconcept relationship.

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow (\Leftrightarrow B_1 \supseteq B_2)$$

These relationships indicate that the extent of the subconcept is contained in the extension of the superconcept, and the subconcept intention contains the superconcept intention.

Figure 1 shows the structure of the concept lattice extracted from the formal context in Table 1. Nodes with blue and black colors mean that there are objects and attributes related to it. Nodes with black half means that there are only objects related to it. Although FCA is an important formalism for the representation of knowledge, it involves complex processes that demand a high computational cost. There are several approaches dedicated to tackle the problem of complexity in the process of generating the concept lattice. The work presented here consists of studying one of these approaches, based on clustering algorithms.

2.2. Fuzzy C-Means X Possibilistic Fuzzy C-Means

In this work, two fuzzy clustering algorithms were used: FCM [Bezdek 1981] and PFCM [Pal et al. 2005]. The PFCM algorithm joins the FCM algorithm and the Possibilistic C Means(PCM) algorithm [Krishnapuram and Keller 1993], with the goal of getting a clustering algorithms with the good characteristics of both.

FCM algorithm is an extension of the K-Means algorithm whose purpose is to find clusters for a set of data. FCM allows data to belong to more than one cluster with a certain degree of membership. The Fuzzy C-Means is a probabilistic method, so the sum of the membership of each object in all clusters must be 1. FCM algorithm finds fuzzy clusters for a data set minimizing an objective function relative to the distances between each data point and the centers of the clusters to which the data belongs with degree of membership $u_{ij} \in [0, 1]$. The objective function J_{FCM} is defined in equation (1), where N is the number of objects, k is the number or clusters and D_{ij} is the distance between object i and cluster center j .

$$J_{FCM} = \sum_{j=1}^N \sum_{i=1}^k (u_{ij})^m D_{ij} \quad (1)$$

Given a data matrix X , and fuzzy partition $U = [u_{ij}] N \times k$, the prototypes (cluster centers) of the i th cluster is calculated as follows:

$$v_i = \frac{\sum_{j=1}^N (u_{ij})^m X_j}{\sum_{j=1}^N (u_{ij})^m} \quad (2)$$

During the minimization process of equation 1, the values of matrix U are updated as shown in equation 3.

$$u_{ij} = \left(\sum_{c=1}^k \left(\frac{D_{ij}}{D_{cj}} \right)^{\frac{1}{(m-1)}} \right)^{-1} \quad (3)$$

Different from FCM, PCM is a possibilistic algorithm, what allows that the sum of the membership values of an instance to all clusters be greater than 1.

The main difference between PCM and FCM is the calculation of the typicity value defined in equation 4. This calculation considers the square distance between data point i and cluster j , so that clusters that are closer to object i have high typicity values, while clusters that are more far away from the object have low typicity values.

$$t_{ij} = \frac{1}{1 + \left(\frac{D_{ikA}^2}{\gamma_i} \right)^{\frac{1}{(\eta-1)}}} \quad (4)$$

In equation 4 η is a constant that can be equal to the fuzzyfier constant m and γ_i is a constant that [Krishnapuram et al. 1995] suggest defining by means of equation 5 with $K > 0$, with $K = 1$ being the most common value.

$$\gamma_i = K \frac{\sum_{j=1}^n u_{ij}^\eta D_{ijA}^2}{\sum_{j=1}^n u_{ij}^\eta} \quad (5)$$

The main goal of the possibilistic approach is to avoid outliers, once it enables the identification of the data that have very low membership values to all clusters, which can be irrelevant to the model. However, PCM algorithm is very susceptible to initial prototypes, converging quickly to local minima.

Therefore PFCM tries to join the two algorithms in order to unite their qualities enabling a better clustering. This way, PFCM proposes to use the two membership calculations and joins them to update the prototypes as defined in 6 where a and b are the constants responsible for weighting each algorithms.

$$v_{ij} = \frac{\sum_{j=1}^n (au_{ij}^m + bt_{ij}^n)x_j}{\sum_{j=1}^n (au_{ij}^m + bt_{ij}^n)} \quad (6)$$

In this way it is possible to vary how much each algorithm (fcm or pcm) will influence the result, being that for values $a = 1$ and $b = 0$ the algorithm becomes a FCM, and $a = 0$ and $b = 1$ becomes a PCM. For this work two variations were used, $a = 1$ and $b = 0$ to use the algorithm FCM and $a = 1$ and $b = 1$ to use the standard PFCM. The PCM ($a = 0$ and $b = 1$) was not used due to its poor results using a random start.

The main challenge in clustering tasks is the analysis and definition of the optimal number of clusters. This problem can be addressed using clustering evaluation metrics, like the one presented in the next subsection.

2.3. Fuzzy Silhouette

The Fuzzy Silhouette (FS) [Campello and Hruschka 2006] in an efficient metric used for fuzzy cluster validation. The FS criterion is useful in the process of obtaining the ideal number of cluster to be generated by the algorithms. As usual in the clustering analysis field, the clustering algorithm is executed on the dataset varying the number of clusters. The evaluation metric is applied to each resulting clustering and the number of clusters in the clustering presenting the best(largest) FS value is chosen.

The Fuzzy Silhouette is based on the similarity between the objects of a cluster and the dissimilarity of these objects in relation to the nearest neighbor cluster, allowing to estimate the best number of k clusters. SF is defined as:

$$SF = \frac{\sum_{i=1}^N (\mu_{pi} - \mu_{qi})^\alpha s_i}{\sum_{i=1}^N (\mu_{pi} - \mu_{qi})^\alpha} \quad (7)$$

In equation 1, α is a weighting coefficient, where $\alpha > 0$ (default value is 1). μ_{pi} and μ_{qi} are the first and second highest membership degree of the $i = th$ column of the fuzzy membership matrix respectively. S_i is the silhouette of the i object defined as:

$$s_i = \frac{b_{pi} - a_{pi}}{\max \{b_{pi}, a_{pi}\}} \quad (8)$$

In which $i = 1, \dots, n$ and $p = 1, \dots, k$. a_{pi} is the average distance of the object i in relation to all objects belonging to the same cluster p . b_{pi} is the dissimilarity of the object i to the nearest neighbor cluster, that is, b_{pi} is the minimum value of the average distance d_{qi} of object i to all objects of another cluster q , where $q \neq p$ e $q = 1, \dots, k$. SF Values are in the range $[-1, 1]$, where SF values closer to 1 mean that the assignment of object i to cluster p the best.

3. Reducing the concept lattice using Simplification

As mentioned before, the simplification technique abstracts nonessential differences (according to some criterion) between concepts, objects, or attributes. Several simplification methods are based on clustering algorithms. Clustering can be performed on a set of formal attributes, objects, or concepts.

In [Cheung and Vogel 2005] the objects are clustered based on the technique of Singular Value Decomposition (SVD), which is one of the several matrix decomposition techniques originated in linear algebra capable of projecting a high dimensional array into a low dimensional array. Equivalence classes of objects are constructed, considering that two objects are equivalent if the cosine of the angles is larger than a pre stipulated threshold. The new object is the joining of all the objects of an equivalence class and its attributes are defined by the union of all the attributes of the equivalent objects. The works described in [Gajdoš et al. 2004], [Snasel et al. 2007] and [Codocedo et al. 2011] also adopt the SVD-based reduction approach. A different proposal, described in [Dias and Vieira 2015], uses non-negative matrix factorization to reduce the formal context.

Reduction techniques based on algebraic methods such as SVD have a high computational cost, making a method unviable mainly for large databases. In an attempt to tackle this problem, Kumar and Srinivas (2010) used the FCM algorithm as an alternative to cluster the objects. After the clustering process, the cluster centers are used to reduce the original formal context to a matrix of lower dimension. According to Krupka (2012), the lattice resulting from the context after the transformation using fuzzy clustering is non-homomorphic to the original.

Another important point identified in some of the approaches found in the literature is the incorporation of background knowledge about the problem domain to guide the simplification process. Some of the most representative works on this line are [Belohlavek and Macko 2011], [Cheung and Vogel 2005], [Dias and Vieira 2010], [Dias and Vieira 2013] and [Zhang et al. 2012].

Although the use of FCM to reduce the number of objects as proposed by Kumar and Srinivas (2010) is advantageous for bringing flexibility and lowering the computational cost, the work described in [Kumar and Srinivas 2010] uses a small illustrative dataset and two numbers of fuzzy clusters, without any kind of assessment of the clustering quality. The work presented in this paper adds an assessment process to define the number of clusters, uses two different clustering algorithms and a number of real datasets.

4. Proposed Method

The method described here is based on the one proposed in [Kumar and Srinivas 2010]. The authors justify the use of FCM saying that some previous studies show that projecting a matrix on the centroids obtained by FCM clustering results a better approximation than K-Means clustering. However, the proposal described in [Kumar and Srinivas 2010] is experimented on very small datasets with two numbers of clusters and compared to the SVD-based method. The novelty of our proposal relies on the evaluation of the best number of groups for each dataset and on a more comprehensive set of experiments and analysis with two different types of fuzzy clustering algorithms. The proposed method comprises the following steps:

1. *Transformation of the data set into a formal context*: The data must be organized in a binary matrix form, where the objects represent each row of the context and the attributes represent each column and the matrix entries define the attributes possessed by the objects.
2. *Formal context clustering and validation of results*
 - 2.1. *Formal context clustering*: The formal context is clustered several times with different numbers of clusters, using the fuzzy clustering algorithms.
 - 2.2. *Results evaluation with Fuzzy Silhouette*: Each one of the clustering obtained in step 2.1 is evaluated with FS and the one with the best (highest) value is selected.
3. *Reduction of formal context based on clustering*: The original formal context is reduced through a process of projection using the clusters centers obtained in the clustering select before and transformed to a matrix with the same structure as the formal context, that is, a matrix composed of objects (lines), attributes (Columns) and the relationships (object-attribute). This transformation takes place through the Decomposition of Concepts [Dhillon and Modha 2001], in which the final result is a matrix approximation D_k . C_k is the centroid matrix resulting from the fuzzy cluster, called matrix of concepts $C_k = [v_1, v_2, \dots, v_k]$ and A is the original formal context.

$$D_k = C_k Z^* \quad (9)$$

$$Z^* = (C_k^t C_k)^{-1} C_k^t A \quad (10)$$

4. *Transformation of the resulting matrix to a binary matrix*: The matrix D_k must be transformed into a binary matrix, returning to be a formal context. The transformation occurs using a pre-defined threshold value, such that matrix values below the threshold become 0 and matrix values greater than the threshold become 1.
5. *Generation of the concepts lattices*: Finally, the concept lattice is generated using one of the well known algorithms for this purpose.

5. Experiments and Results

The datasets used in the experiments (Table 2) contain documents from different sources and have undergone a previous preprocessing step with the objective of reducing the dimensionality, considering that usually documents are represented in very high-dimensional and sparse matrices. In order to do so, the documents sets were clustered and descriptors of documents for each cluster were extracted [Carvalho et al. 2016]. After that, a new "documents X descriptors" matrix is obtained such that the matrix elements represent the frequency of occurrence of each descriptor in each document. The Iaaarticles database is composed of IEEE repository articles collected and manually labeled. The NewYorkTimes database consists of reports from the New York Times site. The Opinosis database contains documents composed of customer comments about the characteristics of some products. Customer's feedbacks were obtained from Tripadvisor.com, where consumers comment on hotels, Amazon.com, where consumers comment on car rentals and Edmunds.com, where consumers comment on electronics. The Newsgroup base originally composed of document collections has 2000 documents approximately divided into 20 categories. For this work, the science category was chosen.

Table 2. Datasets with their number of objects, attributes, concepts and edges of the concept lattice

Datasets	Objects	Attributes	Concepts	Edges
Iaarticles	40	60	23172	114168
Opinosis	51	60	268	676
Reuters	452	120	237160	1122515
NewYorkTimes	18	100	160	436
NewsGroup	400	80	291872	1585123

Table 3. Fuzzy Silhouette (SF) values for the FCM algorithm

Datasets	k=3	k=4	k=5	k=7	k=9
Iaarticles	0.305	0.236	0.219	0.212	0.192
Opinosis	0.470	0.345	0.325	0.346	0.339
Reuters	-	-	-	-	-
NewYorkTimes	0.327	0.255	0.214	0.198	0.142
NewsGroup	0.047	-0.120	0.001	-	-

Due to the high dimensionality of the Reuters and Newsgroup databases, the scope of the experiments was limited by choosing a smaller number of objects and attributes. For the Reuters dataset 452 objects and 120 attributes were selected. For the Newsgroup database the first 400 objects and the same number of attributes (80 attributes) were kept.

All datasets were transformed into binary matrices using a threshold value, so that frequency values above the threshold became 1 and the others became zero. Table 2 shows the number of objects, attributes, concepts and edges for each dataset after the application of FCA algorithms to extract the concept lattice, before the application of the reduction method proposed here.

The numbers of clusters defined to apply the method proposed were 3, 4, 5, 7 and 9. Numbers 5 and 9 were chosen for being the ones used in the work by [Kumar and Srinivas 2010]. The other ones were included to allow a significative variety of results in terms of number of clusters. The fuzzification parameter used was $m = 1.2$ for both algorithms, and the value of $\gamma = 1.2$ was defined for the PFCM algorithm. Table 3 presents the Fuzzy Silhouette values for the clustering results of all k values using FCM algorithm for each dataset. Table 4 shows the same values for the PFCM algorithm. In some cases it was not possible to find a value for SF, because the data were overlapping, not finding clusters well separated. In the FCM algorithm, SF values were not found for Reuters datasets for all clusters and NewsGroup for 7 and 9 clusters. In the PFCM algorithm no SF values were found for 9 clusters in the Datasets Iaarticles and Opinosis. In the NewYorkTimes dataset no SF values were found for 4, 5, 7 and 9 clusters. The cluster centers of the clustering with the highest FS value were chosen to generate the reduced formal context. The highest FS values were for $k = 3$ for both algorithms in all databases.

After the projection of the original context using the cluster centers of the selected clustering to a lower dimensional matrix, it is transformed into a binary matrix. In order to do this transformation, a threshold value must be defined. In this work, the threshold

Table 4. Fuzzy Silhouette (SF) values for the PFCM algorithm

Datasets	k=3	k=4	k=5	k=7	k=9
Iaarticles	0.067	-0.0002	-0.005	0.036	-
Opinosis	0.447	0.160	0.143	0.231	-
Reuters	0.101	0.064	0.001	-0.042	-0.006
NewYorkTimes	0.073	-	-	-	-
NewsGroup	0.057	0.002	0.001	-0.048	-0.039

Table 5. Number Concepts and edges in the concept lattice generated after the fuzzy clustering with FCM and PFCM algorithms.

Datasets	FCM			PFCM		
	Objects	Concepts	Edges	Objects	Concepts	Edges
Iaarticles	40	1247	3974	40	1400	4594
Opinosis	51	140	287	51	211	498
Reuters	452	-	-	452	13228	52251
NewYorkTimes	18	152	378	18	99	214
NewsGroup	400	36436	163116	400	36436	163116

value was defined as the arithmetic mean of the values obtained with the matrix approximation. This definition criterion was selected to avoid values that are too low or too high, since very high threshold values result in a smaller concept lattice, although with greater loss of information and, on the other side, very low threshold values does not lead to an expressive reduction on the formal context.

Table 5 presents the results obtained with the generation of the concept lattice after clustering of the datasets, concerning the number of concepts and edges generated. In this work the concepts have been extracted using ConExp tool [Yevtushenko 2000].

By comparing the results between the two algorithms, it is possible to verify that for a databases with a smaller number of objects and a smaller number of attributes, the FCM algorithm generated fewer concepts and edges. In the databases where the number of objects and the number of attributes is high, the PFCM algorithm generated fewer concepts and fewer edges.

6. Final Considerations

This paper presents a method to reduce the lattice of concepts, based on the simplification technique, in which the reduction occurs using a clustering algorithm on the original formal context. Fuzzy clustering algorithms were chosen for this task because, according to previous experiments it is more flexible than the traditional algorithms. The two fuzzy clustering algorithms used in the experiments have different characteristics, where one is probabilistic (FCM) and the other is the junction of two probabilistic and possibilistic algorithms (PFCM). The experiments demonstrated that for databases with a smaller number of objects and attributes, the FCM obtained a smaller number of concepts and edges. For larger databases, the PFCM obtained the lowest number of concepts and edges.

For future work, we plan to investigate the behavior of the method proposed on different types of datasets and also to analyze the impact of the reduction by means of

more precise measures.

Acknowledgment

The authors would like to thank CAPES for its support.

References

- [Belohlavek and Macko 2011] Belohlavek, R. and Macko, J. (2011). *Selecting Important Concepts Using Weights*, pages 65–80. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Bezdek 1981] Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.
- [Campello and Hruschka 2006] Campello, R. and Hruschka, E. (2006). A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems*, page 157:2858 â 2875.
- [Carvalho et al. 2016] Carvalho, N. V., Rezende, S. O., Camargo, H. A., and Nogueira, T. M. (2016). Flexible document organization by mixing fuzzy and possibilistic clustering algorithms. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 790–797.
- [Cheung and Vogel 2005] Cheung, K. S. and Vogel, D. (2005). Complexity reduction in lattice-based information retrieval. *Information retrieval*, 8(2):285–299.
- [Codocedo et al. 2011] Codocedo, V., Taramasco, C., and Astudillo, H. (2011). Cheating to achieve formal concept analysis over a large formal context. In *The Eighth International Conference on Concept Lattices and their Applications-CLA 2011*, pages 349–362.
- [Dhillon and Modha 2001] Dhillon, I. S. and Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Mach. Learn.*, 42(1-2):143–175.
- [Dias and Vieira 2010] Dias, S. M. and Vieira, N. J. (2010). Reducing the size of concept lattices: The jbos approach. In *Proc. CLA*, pages 80–91.
- [Dias and Vieira 2013] Dias, S. M. and Vieira, N. J. (2013). Applying the jbos reduction method for relevant knowledge extraction. *Expert Systems with Applications*, 40(5):1880–1887.
- [Dias and Vieira 2015] Dias, S. M. and Vieira, N. J. (2015). Concept lattices reduction: definition, analysis and classification. *Expert Systems with Applications*, 42(20):7084–7097.
- [Gajdoš et al. 2004] Gajdoš, P., Moravec, P., and Snášel, V. (2004). Concept lattice generation by singular value decomposition. In *CLA*, volume 2004, pages 13–22. Citeseer.
- [Ganter and Wille 1999] Ganter, B. and Wille, R. (1999). *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin.
- [Krishnapuram et al. 1995] Krishnapuram, R., Frigui, H., and Nasraoui, O. (1995). Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation. ii. *IEEE Transactions on Fuzzy Systems*, 3(1):44–60.

- [Krishnapuram and Keller 1993] Krishnapuram, R. and Keller, J. M. (1993). A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(2):98–110.
- [Kumar and Srinivas 2010] Kumar, C. A. and Srinivas, S. (2010). Concept lattice reduction using fuzzy k-means clustering. *Expert systems with applications*, 37(3):2696–2704.
- [Li et al. 2011] Li, J., Mei, C., and Lv, Y. (2011). A heuristic knowledge-reduction method for decision formal contexts. *Computers & Mathematics with Applications*, 61(4):1096–1106.
- [Maio et al. 2009] Maio, C. D., Fenza, G., Loia, V., and Senatore, S. (2009). Towards an automatic fuzzy ontology generation. In *2009 IEEE International Conference on Fuzzy Systems*, pages 1044–1049.
- [Medina 2012] Medina, J. (2012). Relating attribute reduction in formal, object-oriented and property-oriented concept lattices. *Computers & Mathematics with Applications*, 64(6):1992–2002.
- [Pal et al. 2005] Pal, N. R., Pal, K., Keller, J. M., and Bezdek, J. C. (2005). A possibilistic fuzzy c-means clustering algorithm. *IEEE transactions on fuzzy systems*, 13(4):517–530.
- [Qadi et al. 2010] Qadi, A. E., Aboutajdine, D., and Ennouary, Y. (2010). Formal concept analysis for information retrieval. *CoRR*, abs/1003.1494.
- [Snasel et al. 2007] Snasel, V., Abdulla, H. M. D., and Polovincak, M. (2007). Behavior of the concept lattice reduction to visualizing data after using matrix decompositions. In *Innovations in Information Technology, 2007. IIT'07. 4th International Conference on*, pages 392–396. IEEE.
- [Wang and Zhang 2010] Wang, X. and Zhang, W. (2010). Attribute reduction in concept lattices based on deletion transformations. In *2010 Sixth International Conference on Natural Computation*, volume 4, pages 2065–2069.
- [Yevtushenko 2000] Yevtushenko, S. A. (2000). System of data analysis concept explorer. *Proceedings of the 7th National Conference on Artificial Intelligence KII-2000*, pages 127–134.
- [Zhang et al. 2012] Zhang, S., Guo, P., Zhang, J., Wang, X., and Pedrycz, W. (2012). A completeness analysis of frequent weighted concept lattices and their algebraic properties. *Data & Knowledge Engineering*, 81:104–117.

Evolução Diferencial e Busca local para a Predição da Estrutura de Proteínas

Lucas C. Siqueira¹, Richard A. Gonçalves¹, Carolina P. Almeida¹,
Myriam R. Delgado², Sandra M. Venske¹

¹Universidade Estadual do Centro-Oeste - UNICENTRO
Departamento de Ciência da Computação
Guarapuava, PR, Brasil

²Universidade Tecnológica Federal do Paraná - UTFPR
CPGEI/DAINF
Curitiba, PR, Brasil

llucas.csiqueira@hotmail.com, myriamdelg@utfpr.edu.br

{richard, carol, ssvenske}@unicentro.br

Abstract. *Protein Structure Prediction (PSP) is a Bioinformatics problem in which the goal is to find the tertiary structure of proteins that provides the minimal energy. The approximate search strategy is adequate for this investigation, since the conformational space for this problem is very large. This feature encourages research using evolutionary algorithm approaches. In this work we use evolutionary approach called Differential Evolution to treat the PSP problem. We test two Quasi-Simplex techniques during the optimization process. The results were analyzed with statistical tests.*

Resumo. *A predição da estrutura de proteínas (PSP, do inglês Protein Structure Prediction) é um problema da Bioinformática cujo objetivo é encontrar a estrutura terciária de uma proteína com energia mínima. A estratégia de busca aproximada é adequada para esta investigação, já que o espaço conformacional para o referido problema é muito extenso. Esta característica incentiva a pesquisa utilizando abordagens de algoritmos evolucionários. Neste trabalho propôs-se a investigação da abordagem evolucionária chamada Evolução Diferencial (ED) para tratamento do problema PSP, testando duas técnicas de busca local baseadas no Quasi-Simplex no processo de otimização. Os resultados foram analisados utilizando testes estatísticos.*

1. Introdução

A proteína é uma molécula formada por uma grande cadeia de aminoácidos, sendo o alicerce de uma célula. A proteína realiza praticamente todas as funções biológicas de uma célula, como replicação de DNA, formação das estruturas do citoesqueleto e transporte de oxigênio ao redor dos corpos de organismos multicelulares para a conversão de uma molécula em outra. Existem milhares de diferentes proteínas, cada qual possui uma sequência única de aminoácidos. Para que uma proteína seja biologicamente funcional, deve dobrar-se em um arranjo tridimensional específico, chamado de conformação (que é a estrutura tridimensional ou terciária) [Lehninger et al. 2011].

A predição da estrutura de proteínas (PSP, do inglês *Protein Structure Prediction*) é um problema que, na abordagem *ab initio*, busca prever a estrutura terciária de uma proteína baseada em sua estrutura primária, sem qualquer outra informação *a priori* [Tramontano 2006]. Este problema, cuja solução ainda está em aberto, faz parte do grupo de problemas NP-difíceis [Fraenkel 1993, Hart and Istrail 1997, Crescenzi et al. 1998].

O objetivo do PSP é encontrar a estrutura terciária com energia mínima. A estrutura nativa de uma proteína está relacionada a sua conformação de menor energia livre sendo que, para se chegar a ela, é necessária uma busca no espaço das conformações com o objetivo de selecionar a estrutura com o menor valor global [Tramontano 2006]. A estratégia de busca aproximada é uma boa alternativa para a investigação neste espaço conformacional tão extenso. Esta característica incentiva a pesquisa utilizando abordagens de algoritmos evolucionários para o problema PSP. Neste trabalho será utilizada a abordagem evolucionária chamada Evolução Diferencial.

Evolução Diferencial (ED) [Storn and Price 1997] é um método computacional evolucionário baseado em um processo que envolve o cálculo de uma diferença ponderada entre dois vetores da população e posteriormente a sua adição a um terceiro vetor, considerando parâmetros como fator de mutação, cruzamento e seleção de estratégias de mutação. A ED tem duas características principais que a diferenciam de outros métodos evolucionários. A primeira é que a ED foi originalmente projetada para ser aplicada a espaços contínuos, caso típico do PSP. Em segundo lugar, informações de distância e direção extraídas da população corrente são empregadas a fim de guiar o processo de busca. A distância é representada pela diferença entre dois indivíduos da população e há um indivíduo selecionado para apontar a direção da evolução. Este indivíduo pode ser aleatório ou escolhido como o melhor da população segundo algum critério.

Com a utilização de um algoritmo de busca local espera-se que a convergência e a diversidade das soluções possam melhorar. Alguns trabalhos relacionados podem ser citados. Em [Prestes et al. 2015] é apresentado um algoritmo combinando a busca local *Quasi-Simplex* ao algoritmo MOEA/D, analisando o desempenho do algoritmo proposto em instâncias *benchmark*. Em [Zhang et al. 2010] a técnica *Quasi-Simplex* é aplicada em um algoritmo evolutivo de busca paralela para otimização de função restrita não-linear. Em [Zhang and Lu 2006] é proposto um novo operador de mutação aliado a um algoritmo genético com técnica *Quasi-Simplex*. O algoritmo é testado em funções *benchmark*.

Neste trabalho, propõe-se a aplicação de dois métodos de busca local baseados no *Quasi-Simplex*: um proposto por Lagarias [Lagarias et al. 1998] e outro proposto por Zapotecas-Martínez [Zapotecas-Martínez and Coello 2016]. Cada um dos métodos de busca local é testado separadamente aliado ao algoritmo de evolução diferencial para o problema da predição da estrutura de proteínas. Como teste do algoritmo foi usada a proteína *Met-Enkephalin*, pois esta tem sido escolhida como caso base para diversos estudos para o PSP [Cutello et al. 2006, Liu and Tao 2006, Anile et al. 2007, Cutello et al. 2008, Sudha et al. 2013, Venske et al. 2013].

Este trabalho está organizado da seguinte forma. Na Seção 2 são brevemente discutidos alguns conceitos básicos sobre proteínas, o problema da predição da estrutura de proteínas, a evolução diferencial e os métodos de busca local utilizados neste trabalho. Na Seção 3 é apresentado o algoritmo proposto. A Seção 4 mostra os resultados obtidos

e sua análise. As conclusões do trabalho e os trabalhos futuros são discutidos na Seção 5.

2. Revisão Bibliográfica

2.1. Proteínas

As proteínas são moléculas compostas por aminoácidos, sendo responsáveis por intermediar a maioria dos processos biológicos que ocorrem nas células. Existem 20 tipos de aminoácidos, e estes podem ser considerados o “alfabeto” das proteínas. A combinação desses aminoácidos pode formar milhares de sequências diferentes de proteínas com vários tamanhos. Dentre algumas funções biológicas das proteínas, as principais são: replicação de DNA, transporte de oxigênio, catálise, conversão de uma molécula em outra e formação da estrutura do citoesqueleto das células [Lehninger et al. 2011].

A função molecular das proteínas é determinada pela sua síntese e enovelamento molecular, sendo que o enovelamento é organizado em 4 estruturas hierárquicas: estrutura primária, estrutura secundária, estrutura terciária e estrutura quaternária.

A estrutura primária é formada por uma sequência e um número de aminoácidos diferentes em uma cadeia de proteínas. As estruturas secundárias referem-se ao dobramento de pequenos segmentos contínuos de proteínas em unidades geometricamente ordenadas. O dobramento desses segmentos ocorre por arranjos (folhas)- β e hélices- α .

Na estrutura terciária ocorre a montagem de várias unidades estruturais secundárias em unidades funcionais maiores no espaço em 3D, ou seja, sua estrutura passa a ter forma tridimensional. Na forma tridimensional é que a proteína desenvolve a função que será exercida na célula, sendo possível analisar ou prever essa função.

A estrutura quaternária é constituída pela junção de duas ou mais proteínas na sua forma terciária, ou seja, ela possui mais que uma cadeia polipeptídica. A estrutura da proteína é um pré-requisito chave para se obter a compreensão completa da sua função molecular [Cohen and Kelly 2003]. Quando uma proteína está na sua forma terciária, sua energia potencial livre é mínima [Tramontano 2006].

Além disso, as proteínas têm importante papel na produção de medicamentos. Como a forma tridimensional de uma proteína está diretamente relacionada com a sua função no organismo, a predição de estrutura de proteínas, passou a ser uma das áreas mais pesquisadas da Biologia Molecular [Setubal et al. 1997].

2.2. Problema da Predição da Estrutura de Proteínas

A Predição da Estrutura das proteínas é um problema de grande dimensão e custo computacional da área de Bioinformática. Este é um problema complexo pois faz parte do grupo de problemas NP-difíceis, portanto sua solução ainda está em aberto [Fraenkel 1993, Hart and Istrail 1997, Crescenzi et al. 1998].

O PSP é um problema de minimização, com o objetivo de encontrar a estrutura terciária com a energia mínima dentre um conjunto de conformações de uma proteína. Das milhares de proteínas existentes, apenas se conhece uma pequena parcela das estruturas que as compõem. O conhecimento das estruturas das proteínas é de grande importância, pois com estas informações é possível descobrir a função que a proteína exerce na célula. Além disso, o estudo também pode ajudar os cientistas na descoberta de novas

medicações ou até na cura de doenças como Alzheimer, alguns tipos de câncer, Parkinson, entre outras, já que algumas doenças estão ligadas ao mal funcionamento das proteínas. Existem diversas abordagens que auxiliam no processo de predição da estrutura tridimensional de proteínas. Elas podem ser divididas em dois grupos: baseadas em padrões (homologia e *threading*) e os sem padrões (*ab initio* e *de novo*) [Tramontano 2006]. Neste trabalho foi utilizada a abordagem *ab initio*. Esta representação oferece uma perspectiva mais próxima da realidade do processo de dobramento da estrutura da proteínas, pois estas podem se movimentar livremente no espaço conformacional.

Cada tipo de resíduo de aminoácido tem um número pré-estabelecido de ângulos de torção para alcançar a conformação da proteína. A espinha dorsal de cada resíduo de aminoácido é representada por 3 ângulos: ϕ , ψ , ω . As cadeias laterais são representadas pelos ângulos χ . O número total de ângulos χ depende do tipo de resíduo (conforme apresentado na Tabela 1). Os valores dos ângulos podem variar entre -180° a 180° .

Tabela 1. Número de ângulos χ em cada resíduo de aminoácido.

Resíduo de aminoácido	Número de ângulos χ
GLY, ALA, PRO	cadeia principal
SER, CYS, THR, VAL	χ_1
ILE, LEU, ASP, ASN, HIS, PHE, TYR, TRP	χ_1, χ_2
MET, GLU, GLN	χ_1, χ_2, χ_3
LYS, ARG	$\chi_1, \chi_2, \chi_3, \chi_4$

Há diversos campos de força utilizados para simulação de proteínas. AMBER, CHARMM e GROMOS [Mijajlovic et al. 2010] são exemplos que podem ser utilizados para o cálculo do valor aproximado da energia potencial da conformação de uma proteína. Neste trabalho foi utilizado o modelo CHARMM (v.27), cuja função é apresentada na Equação 1 [Tramontano 2006].

$$Energia_{charmm} = \sum_{lig} K_b(b_C - b_{eq})^2 + \sum_{ang} K_\theta(\theta_C - \theta_{eq})^2 + \sum_{died} K_\phi[1 + \cos(n\Phi_C - \gamma)] + \sum_{nao-lig} \left[\frac{A_{ij}}{r_{ij,C}^{12}} - \frac{B_{ij}}{r_{ij,C}^6} + \frac{q_i q_j}{\epsilon_0 r_{ij,c}} \right] \quad (1)$$

Na Equação 1, K_b é a constante de ligação; K_θ é a constante para ângulos; K_ϕ é a constante para ângulos de torção (diedrais); b é a distância de ligação entre dois átomos; b_{eq} é a distância ideal entre dois átomos; θ é o ângulo da ligação química; θ_{eq} é o ângulo ideal da ligação química; n é o número de mínimos; γ é o deslocamento angular; ϵ é a energia de atração/repulsão de Lennard-Jonnes; A e B são constantes para o potencial de Lennard-Jonnes; q_i e q_j são as cargas de átomos não-ligados; r é a distância entre átomos não-ligados; ϵ_0 é a constante dielétrica.

Para comparar as conformações encontradas pelo algoritmo com a estrutura original da proteína, comumente é utilizado o cálculo de *Root-Mean-Square Deviation* (RMSD), que de maneira geral, mede o grau de semelhança entre duas estruturas [Tramontano 2006]. A fórmula do RMSD é apresentada na Equação 2.

$$RMSD_{C_\alpha}(a, b) = \sqrt{\frac{\sum_{i=1}^{N_a} |\mathbf{r}_{ai} - \mathbf{r}_{bi}|^2}{N_a}} \quad (2)$$

sendo que r_{ai} e r_{bi} são as posições dos átomos i nas estruturas a e b .

2.3. Evolução Diferencial

Algoritmos Evolucionários (AEs) são algoritmos de otimização, baseados em população, que têm como objetivo a resolução de problemas por meios iterativos. Dentre os AEs, destaca-se o algoritmo de Evolução Diferencial [Storn and Price 1997], visando à busca por melhores resultados com uma abordagem um pouco diferente da utilizada por outros AEs e nas estratégias de evolução. Trata-se de um método de busca direta estocástica que surgiu de tentativas de resolver o problema de ajuste polinomial de Chebychev. Kenneth Price introduziu a ideia de usar diferenças de vetores para perturbar a população de vetores (indivíduos) resultando em um método que requer poucas variáveis de controle e possui boa convergência. A ED possui basicamente três parâmetros de controle: tamanho da população (NP), constante de mutação (F) e constante de cruzamento (CR).

Inicialmente, a ED cria uma população inicial aleatória, composta por indivíduos espalhados pelo espaço de busca e avalia estes indivíduos de acordo com uma função de *fitness*. Em seguida, a população é modificada a cada iteração por meio de três operadores da ED sendo eles: mutação, cruzamento e seleção para chegar na próxima geração.

Na mutação são selecionados dois indivíduos aleatórios da população (*vetores diferença*) e em seguida é calculada a diferença ponderada entre esses indivíduos. A esta diferença é aplicado o fator de mutação gerando um indivíduo doador (*vetor doador*) e por fim é feita a adição do indivíduo doador com um indivíduo aleatório da população (*vetor alvo*), assim gerando o indivíduo experimental (*vetor experimental*). Para cada indivíduo $x_{i, Ger}$ da população na geração atual Ger , é gerado o *vetor experimental*, $u_{i, Ger}$, como segue: é selecionado um vetor alvo, $x_{i_1, Ger}$, da população, tal que $i \neq i_1$. Em seguida, são selecionados aleatoriamente dois indivíduos, x_{i_2} e x_{i_3} , tal que i, i_1, i_2 e i_3 são diferentes entre si e $i_1, i_2, i_3 \in (1, NP)$, sendo NP o tamanho da população. O *vetor experimental* é calculado pela perturbação do vetor alvo utilizando estes indivíduos (Equação 3) [Storn and Price 1997]. Está e a estratégia conhecida como $ED/rand/1$ e foi a utilizada neste trabalho.

$$u_{i, Ger} = x_{i_1, Ger} + F(x_{i_2, Ger} - x_{i_3, Ger}) \quad (3)$$

tal que $F \in [0, 1]$ é o fator escalar que controla a amplificação da variação diferencial.

Na operação de cruzamento os componentes do indivíduo experimental são misturados com os componentes de um indivíduo pai (vetor), assim gerando um indivíduo descendente (vetor descendente). A ED possui dois principais tipos de cruzamento sendo eles *cruzamento binomial* e *cruzamento exponencial*. Neste trabalho foi adotado o *cruzamento binomial* descrito pela Equação 4, ficando a estratégia denominada $ED/rand/1/bin$.

$$x'_{i,j} = \begin{cases} u_{i,j}, & \text{se } r_j \leq CR \text{ ou } j = d \\ x_{i,j}, & \text{caso contrário} \end{cases} \quad (4)$$

A Equação 4 gera o vetor descendente x'_i por meio da cópia dos componentes do vetor experimental u_i e do vetor pai x_i , dependendo de um número aleatório $r_j \in [0, 1]$ que é comparado ao parâmetro CR . Devido ao fato que durante o processo de seleção o vetor pai e o vetor descendente irão competir para compor a geração subsequente, é

assegurado que $x'_i \neq x_i$ pela seleção de pelo menos um componente do vetor experimental u_i . Para este propósito a variável $d \in \{1, \dots, D\}$ (D é a dimensão do problema) é escolhida aleatoriamente para cada vetor descendente em cada geração e, desta forma, garante-se que o j -ésimo componente do vetor descendente seja copiado do vetor experimental (somente um componente) [Storn and Price 1997].

No processo de seleção da ED é realizado um teste sobre o indivíduo descendente, em que se verifica se o *fitness* do indivíduo descendente é melhor que o indivíduo pai. Caso essa condição seja verdadeira, o indivíduo descendente será guardado na população para a próxima geração, caso contrário, o indivíduo pai se mantém. Este processo é realizado a fim de determinar qual indivíduo irá para a próxima geração para compor o operador de mutação, produzindo o vetor experimental da próxima geração.

Essas fases de mutação, cruzamento e seleção compõem as estratégias da evolução diferencial. A ED é utilizada no processo de evolução de uma população de proteínas (conformações) a fim de escolher os melhores indivíduos, ou seja, aqueles que tiveram a energia potencial minimizada.

2.4. Busca Local *Quasi-Simplex* proposta por Lagarias

Nesta seção é descrito o método de busca local *Quasi-Simplex* proposto por Lagarias [Lagarias et al. 1998]. Inicialmente, os melhores indivíduos da população da ED são escolhidos para serem utilizados pelo método. Então, um indivíduo chamado *centroide* dos n melhores indivíduos é calculado conforme a Equação 5.

$$X_c = \left(\sum_{i=1}^n X_i \right) / (n) \quad (5)$$

sendo que X_c é o centroide e X_i é i -ésima melhor solução da população. Na sequência, são aplicadas transformações ao conjunto de indivíduos. A intenção das transformações é aprimorar esses indivíduos. As transformações são como descrito a seguir:

- **Reflexão:** do pior (X_w) e do melhor (X_b) indivíduos em relação ao centroide (Equação 6 e 7, respectivamente).

$$X_{rw} = X_w + (X_w - X_c) \quad (6)$$

$$X_{rb} = X_b + (X_c - X_b) \quad (7)$$

sendo que X_{rw} é a reflexão da pior solução e X_{rb} é a reflexão da melhor solução.

- **Expansão:** corresponde a uma grande reflexão do centroide. Há dois tipos de expansão: expansão da pior e da melhor solução. As equações de expansão são dadas por 8 e 9.

$$X_{ew} = X_w + (1.0 + \alpha)(X_w - X_c) \quad (8)$$

$$X_{eb} = X_b + (1.0 + \alpha)(X_c - X_b) \quad (9)$$

sendo que X_{ew} e X_{eb} correspondem a expansão da pior e da melhor solução, respectivamente, e α é um número aleatório uniforme no intervalo (0,1).

- **Compressão:** cria um ponto entre a pior solução e o centroide (X_{cw} - Equação 10) ou entre a melhor solução e o centroide (X_{cb} - Equação 11).

$$X_{cw} = X_w - \alpha(X_w - X_c) \quad (10)$$

$$X_{cb} = X_b - \alpha(X_c - X_b) \quad (11)$$

Todo esse processo é repetido até que um critério de parada seja atendido. Neste trabalho, esse critério corresponde a 50 repetições a cada chamada da busca local. Ao final da aplicação da busca local, as seis melhores soluções obtidas (parâmetro fixo) são retornadas e introduzidas na população, substituindo as piores soluções daquela geração.

2.5. Busca local *Quasi-Simplex* proposta por Zapotecas-Martínez

As transformações do *Quasi-Simplex* proposto por Zapotecas-Martínez [Zapotecas-Martínez and Coello 2016] são compostas por quatro operações: reflexão, expansão, contração externa e interna (Equações 12, 13, 14 e 15, respectivamente).

$$X_r = (1 + \alpha)X_c - \alpha X_p \quad (12)$$

$$X_e = (1 + \alpha\gamma)X_c - \alpha\gamma X_p \quad (13)$$

$$X_{co} = (1 + \alpha\beta)X_c - \alpha\beta X_p \quad (14)$$

$$X_{ci} = (1 - \beta)X_c + \beta X_p \quad (15)$$

sendo que X_r , X_e , X_{co} e X_{ci} são: reflexão, expansão, contração externa e interna, respectivamente; α , β e γ são parâmetros usados nas transformações, com as configurações 1, 2 e 0, 5, respectivamente [Zapotecas-Martínez and Coello 2016]. X_p é o pior indivíduo.

3. Algoritmo Proposto

O Algoritmo 1 descreve o algoritmo baseado em Evolução Diferencial com busca local proposto. O primeiro passo consiste na inicialização aleatória das soluções, respeitando as restrições do problema considerado. Em seguida, o custo de todas as soluções é calculado.

Na sequência o algoritmo entra em seu laço principal, o qual é repetido até que o critério de parada seja satisfeito (neste trabalho foi número máximo de avaliações). No laço principal cada solução passa pelas etapas de mutação, cruzamento e reparação.

A mutação utilizada foi a *ED/rand/1/bin* (Equação 3). O cruzamento adotado foi o binomial (Equação 4). A probabilidade de se escolher um componente do vetor experimental é dada pelo parâmetro *CR*. O cruzamento garante que pelo menos um componente do vetor experimental é escolhido.

Após estas etapas a solução mutada é avaliada e passa pelo processo de seleção. Na seleção, a solução gerada (vetor descendente) substitui a solução corrente se ela possui um menor valor de *fitness*.

A cada *freq_busca* iterações, é aplicado método *Quasi-Simplex* de busca local. O método de busca local utilizado pode ser o proposto por Lagarias (descrito pelo Algoritmo 2) ou por Zapotecas-Martínez (descrito pelo Algoritmo 3). Ele recebe como entrada os 20% melhores indivíduos da população original, que representam os indivíduos (população) da busca local. Após a realização da busca local, seleciona-se o melhor indivíduo encontrado entre as equações do método utilizado, e este substitui o pior indivíduo da população da busca local. Ao fim das *freq_busca* iterações, os 6 melhores indivíduos obtidos na busca local são incluídos na população original.

Algoritmo 1: Algoritmo de ED com busca local

```

1  início
2  Inicializar os parâmetros da ED ( $NP$ ,  $CR$  e  $F$ );
3  Inicializar população de tamanho  $NP$ ;
4  Aplicar reparação, se necessário;
5  Avaliar cada solução da população inicial;
6  repita
7      para cada solução faça
8          início
9              Aplicar a Mutação utilizando  $F$ ;
10             Aplicar o Cruzamento de acordo com  $CR$ ;
11             Aplicar reparação, se necessário;
12             fim
13             Avaliar o custo do indivíduo mutado;
14             Aplicar a seleção;
15             fim
16             se geração  $MOD$   $freq\_busca = 0$  então
17                 Aplicar Quasi-Simplex;
18                 Aplicar reparação, se necessário;
19             fim se
20 até que o critério de parada seja satisfeito;
21 fim

```

Algoritmo 2: Algoritmo *Quasi-Simplex* de busca local proposto por Lagarias

```

1  início
2  para um número de vezes com a população dos melhores faça
3      início
4          Calcular o centroide;
5          Aplicar a Reflexão e avaliar o indivíduo gerado;
6          Aplicar a Expansão e avaliar o indivíduo gerado;
7          Aplicar a Compressão e avaliar o indivíduo gerado;
8          Selecionar o melhor indivíduo dentre os gerados;
9          Substituir o melhor gerado pelo pior indivíduo;
10         fim
11     fim
12 fim

```

4. Resultados Obtidos

A análise dos resultados foi feita baseada em 30 execuções independentes dos algoritmos, considerando a proteína *Met-Enkephalin* (1PLW) como estudo de caso. Os resultados obtidos considerando energia potencial livre, conforme a Equação 1 e em termos da métrica RMSD (Equação 2). A energia é apresentada medida por $kcal\ mol^{-1}$ e o RMSD é medido em termos de Å . Os valores dos parâmetros utilizados são apresentados na Tabela 2.

Um dos parâmetros dos métodos de busca local é com que frequência ela deve ser aplicada durante o processo evolucionário ($freq_busca$). Para os testes, foi avaliada a chamada da busca local a cada 20, 50, 100 e 500 gerações. Na Tabela 3 são apresentados os resultados obtidos com esta abordagem. Na coluna com os valores da frequência da busca utilizada também foi incluído o número total de gerações alcançado. Por exemplo, no primeiro teste do algoritmo para o *Quasi-Simplex*, o processo evolucionário durou

Algoritmo 3: Algoritmo *Quasi-Simplex* de busca local proposto por Zapotecas-Martínez

```

1 início
2   para um número de vezes com a população dos melhores faça
3     início
4       Calcular o centroide;
5       Aplicar a Reflexão e avaliar o indivíduo gerado;
6       Aplicar a Expansão e avaliar o indivíduo gerado;
7       Aplicar a Contração Externa e avaliar o indivíduo gerado;
8       Aplicar a Contração Interna e avaliar o indivíduo gerado;
9       Selecionar o melhor indivíduo dentre os gerados;
10      Substituir o melhor gerado pelo pior indivíduo;
11    fim
12  fim
13 fim
  
```

Tabela 2. Parâmetros para o algoritmo proposto para o PSP.

Parâmetro	Valor	Descrição
NP	100	Tamanho da População.
CR	1,0	Cruzamento.
F	0,5	Fator de Mutação.
MAX-AV	500.000	Número Máximo de Avaliações.
freq_busca	20, 50, 100, 500	Frequência de aplicação da busca local.

4256 gerações, sendo que a cada 20 gerações a busca local foi aplicada.

Tabela 3. Resultados obtidos pelo algoritmo proposto para o PSP considerando a frequência da aplicação da busca local.

	Frequência busca	Menor Energia / RMSD	Média Energia / Desvio Padrão	Mediana Energia	Média RMSD / Desvio Padrão
<i>Lagarias</i>					
	20 (de 4256)	-35,815 / 1,986	-29,014 ± 3,495	-27,433	1,734 ± 0,133
	50 (de 4673)	-35,815 / 1,986	-31,299 ± 3,843	-32,765	1,799 ± 0,152
	100 (de 4830)	-35,814 / 1,987	-33,287 ± 2,698	-33,572	1,795 ± 0,174
	500 (de 4967)	-29,998 / 1,946	-21,863 ± 3,297	-21,597	1,745 ± 0,140
<i>Zapotecas-Martínez</i>					
	20 (de 4443)	-35,815 / 1,987	-28,711 ± 3,770	-27,654	1,709 ± 0,139
	50 (de 4761)	-35,815 / 1,986	-30,272 ± 3,332	-28,683	1,775 ± 0,132
	100 (de 4878)	-35,815 / 1,986	-32,357 ± 3,432	-32,957	1,823 ± 0,161
	500 (de 4976)	-26,520 / 1,805	-20,476 ± 3,012	-20,423	1,718 ± 0,127

O teste de Kruskal-Wallis foi aplicado aos resultados da Tabela 3 sendo considerado um nível de confiança de 95% (i.e., *p-value* abaixo de 0,05). Ele não indicou diferença estatística entre as versões de aplicação da busca local a cada 20, 50 e 100 gerações. O pior resultado de mediana foi obtido pela aplicação da busca local cada 500 gerações, o que fez com que o algoritmo encontrasse os piores valores de energia. Considerando a mediana (e a média) das energias encontradas, há uma pequena vantagem para o uso da busca local a cada 100 gerações, sendo este o valor definido para as comparações seguintes. As conclusões se mantêm para os dois métodos de busca testados.

Na Tabela 4, são apresentados os resultados obtidos comparando os algoritmos que utilizam os dois métodos de busca local com o algoritmo de ED básica. Os dois métodos de busca local superaram a ED básica. O teste de Kruskal-Wallis mostrou que não há diferença entre os dois métodos. Analisando os valores das medianas (e médias) das energias obtidas, o *Quasi-Simplex* proposto por Lagarias teve um resultado melhor do que o proposto por Zapotecas-Martínez. Considerando a métrica RMSD, o algoritmo de ED básica apresentou os melhores resultados.

Tabela 4. Resultados obtidos pelos algoritmos com e sem busca local. Células em destaque representam os melhores valores.

Algoritmo	Menor Energia / RMSD	Média Energia / Desvio Padrão	Mediana Energia	Média RMSD / Desvio Padrão
ED básica	-12,441 / 1,447	-4,656 ± 3,209	-4.898	1,709 ± 0,144
Lagarias	-35,814 / 1,987	-33,287 ± 2,698	-33.573	1,795 ± 0,174
Zapotecas-Martínez	-35,815 / 1,986	-32,357 ± 3,432	-32.957	1,823 ± 0,161

Em geral, os resultados obtidos pelos dois métodos de busca local ficaram bem semelhantes. Contudo, como a versão proposta por Zapotecas-Martínez é menos custosa, pois utiliza um menor número de equações, este foi o método escolhido para ser comparado com a literatura. O resultado desta comparação é apresentado na Tabela 5.

Tabela 5. Resultados obtidos pelo algoritmo proposto e outras abordagens para *Met-Enkephalin*. Células em destaque representam os melhores valores.

Algoritmo	Energia	RMSD _{C_α}
ED + Busca Local Zapotecas-Martínez	-35,81	1,986
PSAGC	-11,10	-
NOMAD-PSP	-30,14	1,550
SaDE/RGA	-12,43	-
IPAES	-20,56	1,740
ADEMO/D	-24,68	1,681

PSAGC (*Parallel Simulated Annealing with Genetic Crossover*) [Liu and Tao 2006] é um algoritmo evolutivo que utiliza *Simulated Annealing* e o campo de força ECEPP/2 para a função de energia. NOMAD-PSP (*Nonlinear Optimization for Mixed Variables and Derivatives algorithm for PSP*) [Nicosia and Stracquandano 2007] é baseado em dois algoritmos evolutivos de busca direta: *Generalized Pattern Search* (GPS) e *Mesh Adaptive Direct Search* (MADS) e utiliza o CHARMM como função de energia. SaDE/RGA [Sudha et al. 2013] utiliza o algoritmo adaptativo SaDE (*Self-adaptive Differential Evolution*) em conjunto com o RGA (*Real-coded Genetic Algorithm*), que é um algoritmo inspirado pela teoria biológica da evolução. Os campos de força para o cálculo das funções de energia utilizados são ECEPP/2 e ECEPP/3. IPAES (*Immune inspired Pareto Archived Evolutionary Strategy*) [Cutello et al. 2006, Cutello et al. 2008] é um algoritmo evolutivo imune com abordagem multiobjetivo, que utiliza o CHARMM. ADEMO/D (*Adaptive Differential Evolution for Multi-objective Problems based on Decomposition*) [Venske et al. 2016] incorpora conceitos de decomposição de problemas e mecanismos de adaptação de estratégias de mutação da ED em uma abordagem multiobjetivo, com a função de energia CHARMM.

O algoritmo de ED com busca local mostrou-se um bom otimizador e obteve o melhor resultado de minimização de energia potencial em relação à literatura. Contudo, considerando os valores de RMSD, o melhor resultado foi alcançado pelo algoritmo NOMAD-PSP, que obteve conformações com a menor distância para a proteína real. A variação do *Quasi-Simplex* obteve o pior resultado dentre os algoritmos comparados.

5. Conclusões

Este trabalho teve como proposta a aplicação de dois métodos de busca local baseados no *Quasi-Simplex*, aliados ao algoritmo de ED para o problema da predição da estrutura de proteínas. Como teste do algoritmo proposto foi utilizada a proteína *Met-Enkephalin*.

Em geral, os resultados apontaram que os dois métodos de busca local possuem a mesma eficácia, comprovados pelo teste de Kruskal-Wallis. Os experimentos mostraram que o algoritmo proposto obteve bons resultados em termos de minimização nos valores de energia potencial em relação à literatura.

Como trabalhos futuros pretende-se testar outras estratégias de Evolução Diferencial, como por exemplo *ED/rand/2* e *ED/current – to – rand/1* com os algoritmos de busca local. O algoritmo também pode ser testado utilizando outras proteínas.

Agradecimentos

Os autores agradecem ao CNPq (Projetos 456.179/2014-3, 483974/2013-7, 309197/2014-7) pelo apoio financeiro.

Referências

- Anile, A. M., Cutello, V., Narzisi, G., Nicosia, G., and Spinella, S. (2007). Determination of protein structure and dynamics combining immune algorithms and pattern search methods. 6:55–72.
- Cohen, F. E. and Kelly, J. W. (2003). Therapeutic approaches to protein-misfolding diseases. *Nature*, 426:905–909.
- Crescenzi, P., Goldman, D., Papadimitriou, C. H., Piccolboni, A., and Yannakakis, M. (1998). On the complexity of protein folding (abstract). In *RECOMB*, pages 61–62.
- Cutello, V., Narzisi, G., and Nicosia, G. (2006). A multi-objective evolutionary approach to the protein structure prediction problem. *J R Soc Interface*, 3(6):139–51.
- Cutello, V., Narzisi, G., and Nicosia, G. (2008). Computational studies of peptide and protein structure prediction problems via multiobjective evolutionary algorithms. In Knowles, J., Corne, D., Deb, K., and Chair, D., editors, *Multiobjective Problem Solving from Nature*, Natural Computing Series, pages 93–114. Springer Berlin Heidelberg.
- Fraenkel, A. S. (1993). Complexity of protein folding. *Bulletin of Mathematical Biology*, 55(6):1199–1210.
- Hart, W. E. and Istrail, S. (1997). Robust proofs of np-hardness for protein folding: General lattices and energy potentials. *Journal of Computational Biology*, 4(1):1–22.
- Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E. (1998). Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147.

- Lehninger, A., Nelson, D., and Cox, M. (2011). *Princípios de bioquímica*. ARTMED.
- Liu, Y.-L. and Tao, L. (2006). An improved parallel simulated annealing algorithm used for protein structure prediction. In *Machine Learning and Cybernetics, 2006 International Conference on*, pages 2335–2338.
- Mijajlovic, M., Biggs, M. J., and Djurdjevic, D. P. (2010). On potential energy models for ea-based ab initio protein structure prediction. *Evol. Comput.*, 18:255–275.
- Nicosia, G. and Stracquadanio, G. (2007). Generalized pattern search and mesh adaptive direct search algorithms for protein structure prediction. In *Proceedings of the 7th International Conference on Algorithms in Bioinformatics, WABI'07*, pages 183–193, Berlin, Heidelberg. Springer-Verlag.
- Prestes, L., Almeida, C., and Gonçalves, R. (2015). Implementation of quasi-simplex local search on moea/d. In *Computational Intelligence (LA-CCI), 2015 Latin America Congress on*, pages 1–6. IEEE.
- Setubal, J. C., Meidanis, J., and Setubal-Meidanis (1997). *Introduction to computational molecular biology*. PWS Pub.
- Storn, R. and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. 11(4):341–359.
- Sudha, S., Baskar, S., and Krishnaswamy, S. (2013). Protein tertiary structure prediction using evolutionary algorithms. *International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)*, 3(3):338–348–595.
- Tramontano, A. (2006). *Protein Structure Prediction: Concepts and Applications*. John Wiley and Sons.
- Venske, S. M., Gonçalves, R. A., Martin, E. B., and Delgado, M. R. (2013). A multiobjective algorithm for protein structure prediction using adaptive differential evolution. In *2013 Brazilian Conference on Intelligent Systems*, pages 263–268.
- Venske, S. M., Gonçalves, R. A., Benelli, E. M., and Delgado, M. R. (2016). Ademo/d: An adaptive differential evolution for protein structure prediction problem. *Expert Systems with Applications*, 56:209 – 226.
- Zapotecas-Martínez, S. and Coello, C. A. C. (2016). Monss: A multi-objective nonlinear simplex search approach. *Engineering Optimization*, 48(1):16–38.
- Zhang, G. and Lu, H. (2006). Hybrid real-coded genetic algorithm with quasi-simplex technique. In *IJCSNS International Journal of Computer Science and Network Security*, pages 246–255.
- Zhang, G. L., Lu, H. Y., and Zhang, G. Q. (2010). A new hybrid evolutionary algorithm with quasi-simplex technique. In *2010 International Conference on Machine Learning and Cybernetics*, volume 4, pages 1811–1816.

Semantic Parsing Natural Language into SPARQL: an LSTM Encoder-Decoder Neural Net Approach

Fabiano Ferreira Luz¹, Marcelo Finger¹

¹Universidade de São Paulo - USP
Instituto de Matemática e Estatística - IME
Departamento de Ciência da Computação
Rua do Matão, 1010 - Cidade Universitária
São Paulo - SP - Brasil - CEP 05508-090

{fluz,mfinger}@ime.usp.br

Abstract. *Semantic parsing is the process of mapping a natural language sentence into a formal representation of its meaning. In this work we use the neural network approach to transform natural language sentence into a query to an ontology database in the SPARQL language. This method does not rely on handcraft-rules, high-quality lexicons, manually-built templates or other handmade complex structures. Our approach is based on vector space model and neural networks. The proposed model is based in two learning steps. The first step generates a vector representation for the sentence in natural language and SPARQL query. The second step uses this vector representation as input to a neural network (LSTM with attention mechanism) to generate a model able to encode natural language and decode SPARQL.*

1. Introduction

Semantic parsing can be defined as the process of mapping natural language sentences into a machine interpreted, formal representation of its meaning. Currently, there are many efforts aimed at transforming human language into a computational representation [Zettlemoyer and Collins 2012, Alshawi et al. 2014, Bowman et al. 2014].

In this work, we are concerned with semantic parsing as the task of transforming natural language into SPARQL queries. In recent years this task has received a lot of attention [Wang et al. 2007, Lehmann and Bühmann 2011, Ferré 2012, Ferré 2017]. This is mainly due to the increase of RDF-based documents available on the web, with public repositories such as DBpedia [Auer et al. 2007], which already provides support for SPARQL queries in its database.

Traditional approaches to translating natural language into SPARQL rely on high-quality lexicons, manually-built templates, and linguistic features which are either domain or representation-specific [Dong and Lapata 2016]. This is a problem since adapting such models to a new domain can be a very laborious task. Because of this, we propose a model based on artificial neural networks that among other advantages, does not rely on handcraft rules, high-quality lexicons, manually-built templates or other handmade complex structures.

In our work, we used an LSTM encoder-decoder model capable of encoding natural language (English) and decoding query language (SPARQL). The first part of our

work is dedicated to finding suitable vector representations for sentences both in natural language and in SPARQL. On the natural language side, we use the vector-based Glove model [Pennington et al. 2014]. For the SPARQL part, we propose a composition of methods to generate the vector representation. In addition, we propose our own way of generating this representation. In the second part of this work, we implement and configure an LSTM encoder-decoder model tailored to the task of translating natural language queries to SPARQL.

The contributions of this paper are the following.

- The creation of a version of the Geo880 dataset with SPARQL queries like a target language.
- The creation of an OWL ontology of the Geo880 domain.
- The development of a novel vector representation for the target language lexicon.
- The adaptation of the encoder-decoder model with neural attention to transform natural language into SPARQL queries.

This work is organized as follows. In Section 2, we provide a background of the elements needed to understand how our model works. In Section 3 we detail our model and the lexical generation procedure. Section 4 is dedicated to discussing experiments and results. In Section 5 we talk more about related work and in the last section we comment on our contribution and future work.

2. Background

In the following sections we describe the neural network architecture encoder-decoder and also discuss the concept of neural attention.

2.1. Recurrent Neural Network

A Recurrent Neural Network (RNN) is a type of artificial neural network where connections between units form a directed cycle. This cycle represents an internal state of the network which allows it to exhibit dynamic temporal behavior. RNNs can use their internal memory to process arbitrary sequences of inputs. An output of a hidden layer h_t of an RNN can be defined as

$$h_t = f(h_{t-1}, x_t), \quad (1)$$

where h_{t-1} , is the value of the hidden layer at time $t - 1$, x_t is the input feature vector and $f(\cdot)$ is a nonlinear function.

It has been noted by [Bengio et al. 1994] that RNNs suffer from the *vanishing gradient problem*, which consists of the exponential decrease that the value of $h_{t'}$ has influence over the value of h_t , $t' < t$, leading to a very short the temporal memory in the network. One solution to this problem was a change in the neuron's nucleus called Long Short Term Memory (LSTM) [Hochreiter and Schmidhuber 1997]. RNN-LSTM has been used successfully in language modeling problems because, so now it can handle long sequences quite well.

2.2. Encoder-Decoder Model

The Encoder-Decoder model, proposed by [Cho et al. 2014], is a neural network architecture that learns to learn the conditional distribution of a conditioning variable-length sequence \mathbf{x} in another variable-length sequence \mathbf{y} . It performs this task by learning how to *encode* a variable-length sequence x_1, \dots, x_T into a fixed-length vector representation c and then to *decode* a given fixed-length vector representation c back into a variable-length sequence y_1, \dots, y_S . The function may be interpreted as the distribution $p(y_1, \dots, y_S | x_1, \dots, x_T)$; the input sequence length T and output one S can be different.

The *encoder* is an RNN that reads each word of an input sequence \mathbf{x} sequentially. As it reads each symbol, the hidden state of the RNN is updated according to equation (1). After reading the end of the sequence (marked with an end sequence symbol), the hidden state of the RNN is summarized. We call this summary c . In order to simplify we can define c as the output h .

The *decoder* is another RNN which is trained to generate the output sequence by predicting the next symbol y_t given the hidden state h_t . However, unlike the RNN described previously, here both y_t and h_t are conditioned to y_{t-1} and the summary c of the input sequence. Thus, the hidden state of the decoder at time t is computed by: $h_t = f(h_{t-1}, y_{t-1}, c)$, and likewise, we can define the conditional distribution of the next symbol by the following equation:

$$P(y_t | y_{t-1}, \dots, y_1, c) = g(h_{t-1}, y_t, c). \quad (2)$$

The activation function g produces valid probabilities by, for example, by computing the *softmax function*. Figure 1 presents an overview of the encoder and decoder scheme.

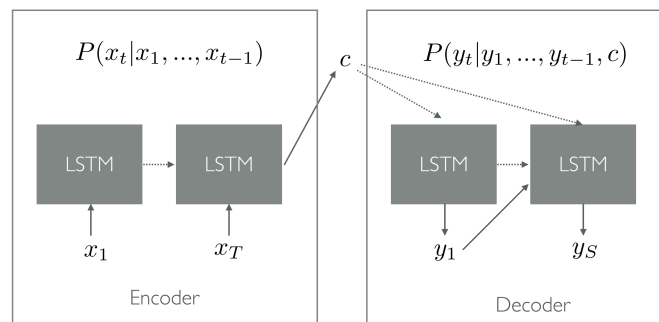


Figure 1. Encoder-Decoder scheme.

The combination of the two described components (Encoder and Decoder) make up the training of the proposed model to maximize the conditional log-likelihood and can be represented by the equation:

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^N \log p_{\theta}(\mathbf{y} | \mathbf{x}), \quad (3)$$

where θ is the set of the model parameters and each pair (\mathbf{x}, \mathbf{y}) is, respectively, an input and output sequence. In our case, we use the vector representation of questions in natural language as input and the SPARQL query as an output. As the output of the decoder, starting from the input, is differentiable, we can use a gradient-based algorithm to estimate the model parameters.

After training the encoder-decoder RNN, the model can be used in two distinct ways. In the first case, we can use the model to generate a target sequence, once the input sequence is provided computing the most probable output given the input. In the second one, the model can be used to evaluate a given pair of input and output sequences by calculating a score (e.g. the probability $p_{\theta}(\mathbf{y}|\mathbf{x})$).

Although sequential models present good results for sequence transformation, they still present a distortion in the sentence mapping task because relationships among sentences are largely organized in terms of latent nested syntactic/semantic structures rather than sequential surface order [Lees and Chomsky 1957, Dyer et al. 2016]. One way to deal with these linguistic properties is adding the mechanism of *neural attention*, as used, for example, in soft alignment [Bahdanau et al. 2015].

2.3. Attention Mechanism in Neural Networks

The model proposed by [Bahdanau et al. 2015] differs from the basic model of encoder-decoder by not attempting to encode a full entry into a fixed-size vector. Instead, it encodes entries into a sequence of vectors and selects a subset of them adaptively while decoding the translation. With this modification, the neural network no longer has the challenge of compressing information of an entire sequence into a fixed-size vector. The new architecture consists of a bidirectional RNN as an encoder and a decoder that emulates searching through a source sentence while decoding a translation.

New Encoder: The proposed encoder in [Bahdanau et al. 2015] does not use a standard RNN described in equation (1), which reads an input sequence \mathbf{x} starting from the first element x_1 to the last x_{T_x} . However, in the proposed scheme, the encoder does not compute only a single summary of the previous words. Instead, for each input word it computes an *annotation* representing both a summary of the previous words and also one for the following ones. Then, an appropriate model to obtain such annotation is a bidirectional RNN [Schuster and Paliwal 1997], which has recently been used successfully in speech recognition [Graves et al. 2013].

A bidirectional RNN is composed of a forward RNN \vec{f} that reads the input sequence as it is ordered from x_1 to x_{T_x} and calculates a sequence of *forward hidden states* $(\vec{h}_1, \dots, \vec{h}_{T_x})$ and the backward RNN \overleftarrow{f} which reads the sequence in reverse order, from x_{T_x} to x_1 , resulting in a sequence of *backward hidden states* $(\overleftarrow{h}_1, \dots, \overleftarrow{h}_{T_x})$.

Thus, an annotation is obtained for each word x_j by concatenating the forward hidden state \vec{h}_j and the backward one \overleftarrow{h}_j ,

$$h_j = \left[\vec{h}_j^T; \overleftarrow{h}_j^T \right]^T. \quad (4)$$

The annotation h_j encodes both preceding and following words. Due to the tendency of RNNs to better represent recent inputs, the annotation h_j is focused on the words around x_j . This sequence of annotations is used by the decoder and the alignment model later to compute the context vector.

New Decoder: In the new model, each conditional probability is defined by: $p(y_i|y_1, \dots, y_{i-1}, x) = g(y_i, s_i, c_i)$, where s_i is an RNN hidden state for time i , computed by:

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \quad (5)$$

Unlike the basic encoder-decoder approach in (2), here the probability is conditioned on a distinct context vector c_i for each target word y_i . The context vector c_i depends on a sequence of *annotations* (h_1, \dots, h_{T_x}) in which an encoder maps the input sentence. Each annotation h_i consists of information about the whole input sequence with a strong focus on the parts surrounding the i -th word of the input sequence.

The context vector c_i is then computed as a weighted sum of these annotations h_i

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \quad (6)$$

The weight α_{ij} of each annotation h_j is computed by

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \quad (7)$$

where $e_{ij} = a(s_{i-1}, h_j)$ is an alignment model that scores how well the input around position j and the output at position i match. The score is based on the RNN hidden state s_{i-1} of (5) and the j -th annotation h_j of the input sentence.

The alignment model a is parametrized as a feedforward neural network which is concomitantly trained with all the other components of the proposed system. The alignment model directly computes a soft alignment, which allows the gradient of the cost function to be backpropagated. This gradient can be used to train the alignment model as well as the whole translation model at the same time.

The use of a weighted sum of all annotations can be interpreted as calculating an *expected annotation*, where the expectation is over possible alignments. Let α_{ij} be a probability that the target word y_i is aligned to, or translated from, a source word x_j . Then, the i -th context vector c_i is the expected annotation over all the annotations with probabilities α_{ij} .

For visual reasons, the alignment matrix of Figure 2 was plotted with colors instead of numerical values. The matrix is an example of the alignment obtained during the training of our model. The darker the cell of the matrix the bigger the correlation between the terms row and column.

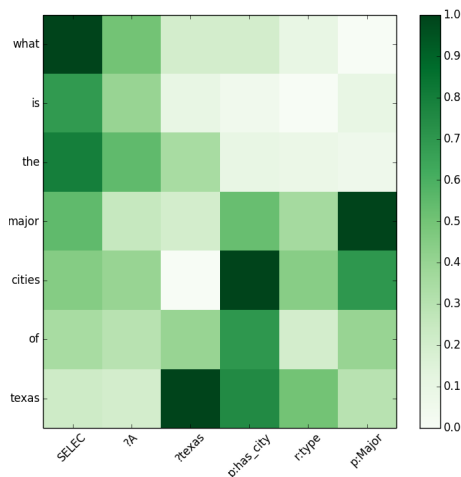


Figure 2. Neural attention matrix example

3. Our Approach

The model described here is based on two approaches in the neural network literature. First, a neural probabilistic language model similar to that of [Pennington et al. 2014] is used to learn a word vector representation, and a LSTM neural network similar to that of [Dong and Lapata 2016] is used to encode natural language sentences and decode SPARQL query. We provided the necessary background on these two components in Section 2.

There are at least two important contributions in our work. The first is the fact that we use the model in question to translate from natural language to SPARQL. Another important contribution is the representation of the lexicon we are proposing. For this representation we use the neural attention mechanism to generate the table that will be used in matching. This section is divided into three parts where, first, we give an overview of the developing model; then we describe the concept of *matching* using neural attention; and in the end we detail how we generate the lexical representation for the target language.

3.1. General overview of the approach

Our approach consists of two learning phases, which can be observed in Figure 3:

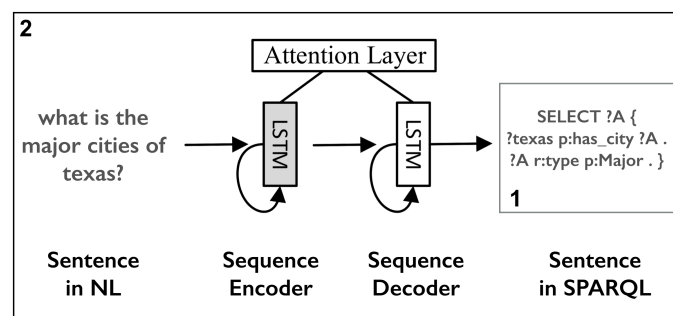


Figure 3. Overview

The following text, we have a detailed description of each step:

1. The first step is to find a good vector representation for the target language lexicon. To represent the source natural language, we employ the method in [Pennington et al. 2014].
2. In the second part, we are concerned with implementing and finding the settings so that our architecture can translate from natural language to SPARQL.

For the training of our model, a dataset of paired sentences is necessary where, on the one hand we have questions in natural language and the other their SPARQL. This dataset is described in Section 4.

3.2. Vocabulary matching using neural attention

In this work we propose a kind of dictionary where for each word of the target language we have another one in the source language that has the *greatest correlation*. Let T be target language vocabulary and S be source language vocabulary. Λ is the total alignment matrix of dimension $|S| \times |T|$ where Λ_{ij} contains the value of correlation between words s_i and t_j . We can define the *greatest correlation* of a word t_j from the target language as the:

$$\max_{i=1..|S|} (\Lambda_{ij})$$

and the word $w_{i^*}, i^* = \operatorname{argmax}(\Lambda_{ij})$, is the word from the source language that possesses the *greatest correlation* with t_j .

We can exemplify the matching procedure with Figure 2 from the previous section. Note that the target word **p:has_city** has the corresponding word **cities**. We can observe this because the darker cell connects the two words. That is, the word **p:has_city** has *greater correlation* with the word **cities**. Based on this table, we have built a translation dictionary from target to source. This dictionary could have several practical uses. In our work, it will be useful in the lexical representation of the target that will be better explained in the next section.

3.3. Lexical representation

To represent the lexicon of natural language, our source language, we use the model developed in [Pennington et al. 2014]. However, in the case of the SPARQL terms we did several experiments to analyze different representations. Next we describe the vector representation generation methods for the lexicon we use.

RANDOM: This representation was generated randomly using a normal distribution as a kernel with values from -1 to 1. This approach was chosen in order to kick off the representation like a baseline and even if it could not capture the relationship between terms, its performance was not so low when compared to more sophisticated approaches.

TF-IDF/PCA: We use the target language sentences and generate a term document matrix. In this matrix, we consider each query a document and each word in the target language a term. After generating the term-document matrix we apply TF-IDF [Aizawa 2003]. At the end we did a size reduction using Principal Component Analysis (PCA) [Jolliffe 2002]. Our reduction was from 880 to 300 dimensions.

W2V10: This approach is based on work by [Mikolov et al. 2010], which proposed the generation of a vector representation of words based on recurrent neural networks, using only text as input. We use the target language sentences as input text to generate a vector representation.

OUR-APP: Our approach is focused on using the same set of vectors to represent both source and target language vocabulary. What we do is a match between the terms of the source language and the target language seeking a match between the two vocabularies. The matching process is possible with the alignment table generated by the neural attention mechanism.

Then, first using a random version to represent the target’s lexicon, we train and generate an alignment table, Figure 2. Then, we use the **Matching** (see section 3.2) to associate the lexicon of the source language with the target language. We use a heuristic to associate all elements of one vocabulary with another. SPARQL words are similar to English words, for example, “SELECT”, “FILTER”, “ORDER”, etc were directly associated with their respective correspondents in English. The other terms such as, subject, predicate and object of the triples, are solved with the matching procedure. Our heuristics proved to be good according to the results. A full description of the heuristics used and how to deal with the generation of non-grammatical (incorrect) SPARQL expressions will be dealt with in future work.

Note that in this work we do not deal with the “out of vocabulary” situation (OOV), nor with lexical disambiguation problem. In fact, the OOV problem is avoided by the word2vec pre-processing, and every word is assumed to be associated to a vector of features, which is comparable to having a controlled vocabulary. Similarly, the lexical ambiguities are assumed to be codified inside the attribute vectors in the pre-processing; furthermore, approximately 1% of the words in the vocabulary are lexically ambiguous, allowing us to safely ignore such effects.

4. Experimental Evaluation

We compared our methodology with related work using the Geo880 dataset. We use only one dataset because we needed to create a SPARQL corpus for each domain and it is very laborious. The test with different datasets remained as future work.

In this section, first, we define our metrics and then talk about an adaptation of the dataset to the SPARQL query. We also talk about syntactic errors, neural network settings and finally comment on comparisons of our work with other different approaches. Here we use two metrics to evaluate our approach:

$$\text{Accuracy} = \frac{\# \text{ of correctly translated queries}}{\# \text{ total of queries}}$$

$$\text{Syntactical Errors} = \frac{\# \text{ of queries with syntactical errors}}{\# \text{ total of queries}}$$

4.1. Dataset

Our experiments were conducted using one traditional dataset: Geo880, a set of 880 queries to a database of U.S. geography. The data were originally annotated with Prolog style semantics which we manually converted to equivalent statements in SPARQL

queries. On the official web page of Geo880 dataset of the University of Austin in Texas¹ we found some files. Using two files, the **geobase** with assertions of the dataset and the **geoquery880** file, containing questions directed at this domain. First we created an OWL ontology based on the geodata file and then, for each query in natural language on geoquery880 file, we wrote a corresponding SPARQL query. Both the ontology and the set of questions can be found in our repository <https://github.com/mllovers/geo880-sparql>.

4.2. Syntactical Errors

We call a *syntactical error* when a generated SPARQL statement cannot be processed by Protégé² due to syntactical formation³. In general, there are several syntactical errors that can be generated, such as not closing any brackets or even trying to apply a function to a variable of the wrong type.

4.3. Settings

Finding better parameters and hyper-parameters for neural networks is always a very costly task. The first rounds with the neural network, the pre-tests, served to find the best parameters for the neural network. In all, the pre-tests lasted more than two months.

- Learning rate: During the pre-test, we use two different learning rates. The first, we maintained the learning rate at 0.5. The second was started learning rate at 0.9 and decreasing 0.9 of it in each iteration (epoch). The best results obtained are used in the second case;
- Epoch: In the tests we chose to use 100 epochs, as this was the best result found in our pre-tests for hyperparameters;
- Hidden dimension: With regard to the number of hidden layers, we used a series of pre-tests with 100, 200 and 400 hidden layers. We continued testing with the 4 different dimensions;
- Input dimension: We used three different input dimensions, 100, 200 and 300 but the best results were obtained with vectors of size 300.

4.4. Results

In Table 1 we show the experiments performed without the mechanism of neural attention. All experiments were performed with the Geo880 dataset. We used 10-fold crossvalidation.

LSTM Encoder-decoder - Geo880 Dataset			
Hidden Dim.	Method	Synt. error	Accuracy
100	RANDOM	14.77	30.68
	TF-IDF/PCA	13.64	31.89
	W2V10	14.77	28.40
	OUR-APP	11.36	38.63
200	RANDOM	11.36	36.36
	TF-IDF/PCA	10.23	38.67
	W2V10	11.36	32.95
	OUR-APP	07.95	54.55
400	RANDOM	09.09	51.14
	TF-IDF/PCA	07.95	54.55
	W2V10	10.23	42.04
	OUR-APP	06.81	64.77

Table 1. Without neural attention

LSTM Encoder-Decoder - Geo880 Dataset			
Hidden Dim.	Method	Synt. Error	Accuracy
100	RANDOM	12.50	37.50
	TF-IDF/PCA	11.36	40.91
	W2V10	12.50	35.23
	OUR-APP	0	54.55
200	RANDOM	11.36	47.73
	TF-IDF/PCA	10.22	51.14
	W2V10	11.36	45.45
	OUR-APP	09.09	67.04
400	RANDOM	07.95	62.50
	TF-IDF/PCA	06.81	64.77
	W2V10	09.09	60.23
	OUR-APP	05.68	78.40

Table 2. With neural attention

¹<http://www.cs.utexas.edu/users/ml/nldata/geoquery.html>

²<http://protege.stanford.edu/>

³Another option would be to use a grammar that describes SPARQL, then use it as parameter of the parser algorithm. However a rejection by the SPARQL interpreter is a cheaper option found.

In Table 2 the tests were run on the LSTM encoder-decoder architecture with attention mechanism. In next table we show examples of inputs and outputs generated by our model.

Input	Output (Without IRI Prefix to simplify)
how many rivers are there in idaho ?	SELECT (COUNT(?A) AS ?QTD) { ?idaho p:river ?A FILTER (regex(str(?idaho), "idaho", "i")) . }
show major cities in colorado ?	SELECT ?A { ?colorado p:city ?A FILTER (regex(str(?colorado), "colorado", "i")) . ?A r:type p:Major . }
what are the cities of the state with the highest point ?	SELECT ?B { ?A p:city ?B . { SELECT ?A { ?A r:type p:State . ?A p:highest_point ?B . ?B p:height ?height . } ORDER BY DESC(?height) LIMIT 1 } }

Regarding the task of transforming the natural language into SPARQL, we compared our work with [AlAgha 2015] and [Kaufmann et al. 2006]. In the first paper, the authors also use the Geo880 dataset and through linguistic analysis identify elements of natural language and generate RDF triples. In the second paper, the main strategy of the authors was to try to associate triples of natural language with RDF triples. As can be seen in the Table 3, we obtained better results in the tests with dataset Geo880.

	Accuracy
[AlAgha 2015]	58.61
Querix [Kaufmann et al. 2006]	77.67
Our method	78.40

Table 3. Natural Language to SPARQL comparisons

Although we mention the work [Wang et al. 2007], that also makes use of the Geo880 dataset, we do not make the comparison with it because it does not use the original set of Geo880 queries in their tests. For the same reason, we also disregard one of the results of Querix [Kaufmann et al. 2006]. With respect to the **syntactical errors**, we can see in our tests that the better the model in general, the lower the error rate of syntax. We also propose as future work to develop a model that the generated sentence has syntactic guarantee.

5. Related Work

In this section we discuss three related works that are related to different aspects of our model. The first, [Dong and Lapata 2016], is related to the task of mapping natural language sentences to the logical form. The second [AlAgha 2015] and third [Kaufmann et al. 2006], are related with the task of translating natural language for SPARQL.

[Dong and Lapata 2016] present a general method based on an attention-enhanced sequence- to-sequence model. They encode input sentences into vector representations using recurrent neural networks, and generate their logical forms by conditioning the output on the encoding vectors. The model is trained in an end-to-end fashion to maximize the likelihood of target logical forms given the natural language inputs. Although they do not deal with SPARQL, their approach uses a neural network attention-based structure similar to ours.

The work in [AlAgha 2015] associates phrases in natural language with RDF triples, as in our approach. Through a linguistic analysis, their model extracts relations and associates them to triples. They also generate SPARQL scripts, however using first an intermediate format. Then, with the help of an ontology, a SPARQL query is generated after identifying the targets and modifiers of the query. That is developed using Arabic as the source language, however a comparison with ours can be made as both works also use information extraction from a syntactic tree.

The Querix [Kaufmann et al. 2006] employs a statistical approach. Given a query, the system consists of parsing, removing important elements and then looking for triples that are related to the elements of the query. Querix works as a component and can be adapted in any application. It is based on clarification of dialogues, so when there is ambiguity the system asks the user to decide. All these works were evaluated using the GEO 880 dataset.

6. Discussion and Future Work

The purpose of this work was to explore artificial neural network resources in the development of a model that may be able to translate from natural language to SPARQL. The choice of the OWL Ontology and the SPARQL language as the target language is due to the fact that we are interested in practical applications. Among the advantages of using artificial neural networks, we can highlight the fact that we do not need linguistic knowledge nor do we depend on the development of complex grammars.

In addition to dealing with SPARQL, we propose in this work a representation of the target language lexicon that according to the results was a good approach. This representation is only possible because we use the concept of Matching of terms oriented by the alignment table that is generated by the mechanism of neural attention. Moreover, we can highlight as main future works: To perform tests with different datasets and to guarantee of correct syntax in query generation.

Acknowledgment

This work was developed with the support of the National Council for Scientific and Technological Development (CNPq). Marcelo Finger was partly supported by Fapesp projects 2015/21880-4 and 2014/12236-1 and CNPq grant PQ 306582/2014-7.

References

- [Aizawa 2003] Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.
- [AlAgha 2015] AlAgha, I. (2015). Using linguistic analysis to translate arabic natural language queries to SPARQL. *CoRR*, abs/1508.01447.
- [Alshawi et al. 2014] Alshawi, H., Chang, P.-C., and Ringgaard, M. (2014). Deterministic statistical mapping of sentences to underspecified semantics. In *Computing Meaning*, pages 13–25. Springer.
- [Auer et al. 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735.
- [Bahdanau et al. 2015] Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the ICLR*.
- [Bengio et al. 1994] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.
- [Bowman et al. 2014] Bowman, S. R., Potts, C., and Manning, C. D. (2014). Recursive neural networks for learning logical semantics. *CoRR*, abs/1406.1827, 5.

- [Cho et al. 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *In Proceedings of the 2014 EMNLP*, pages 1724–1734.
- [Dong and Lapata 2016] Dong, L. and Lapata, M. (2016). Language to logical form with neural attention. *CoRR*, abs/1601.01280.
- [Dyer et al. 2016] Dyer, C., Kuncoro, A., Ballesteros, M., and Smith, N. A. (2016). Recurrent neural network grammars. *CoRR*, abs/1602.07776.
- [Ferré 2012] Ferré, S. (2012). Squall: A controlled natural language for querying and updating rdf graphs. In *International Workshop on Controlled Natural Language*, pages 11–25. Springer.
- [Ferré 2017] Ferré, S. (2017). Sparklis: an expressive query builder for sparql endpoints with guidance in natural language. *Semantic Web*, 8(3):405–418.
- [Graves et al. 2013] Graves, A., Jaitly, N., and Mohamed, A.-r. (2013). Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE.
- [Hochreiter and Schmidhuber 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [Jolliffe 2002] Jolliffe, I. (2002). *Principal component analysis*. Wiley Online Library.
- [Kaufmann et al. 2006] Kaufmann, E., Bernstein, A., and Zumstein, R. (2006). Querix: A natural language interface to query ontologies based on clarification dialogs. In *5th International Semantic Web Conference (ISWC 2006)*, pages 980–981. Springer.
- [Lees and Chomsky 1957] Lees, R. B. and Chomsky, N. (1957). Syntactic structures. *Language*, 33(3 Part 1):375–408.
- [Lehmann and Bühmann 2011] Lehmann, J. and Bühmann, L. (2011). Autosparql: Let users query your knowledge base. *The Semantic Web: Research and Applications*, pages 63–79.
- [Mikolov et al. 2010] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech*, volume 2, page 3.
- [Pennington et al. 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.
- [Schuster and Paliwal 1997] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11):2673–2681.
- [Wang et al. 2007] Wang, C., Xiong, M., Zhou, Q., and Yu, Y. (2007). Panto: A portable natural language interface to ontologies. In *European Semantic Web Conference*, pages 473–487. Springer.
- [Zettlemoyer and Collins 2012] Zettlemoyer, L. S. and Collins, M. (2012). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *CoRR*, abs/1207.1420.

Uma Hiper-Heurística para o Flowshop Multiobjetivo

Geovani Ferreira Antunes¹, Richard Aderbal Gonçalves¹, Sandra Mara Venske¹, Carolina Paula de Almeida¹

¹Departamento de Ciência da Computação – Universidade Estadual do Centro-Oeste (UNICENTRO)
Rua Simeão Camargo Varela de Sá, 03 - Vila Carli – 85040-080
Guarapuava – PR – Brasil

geovanyantunes@hotmail.com, {richard, ssvenske, carol}@unicentro.br

Abstract. *The Flowshop is a sequencing problem used in the management of the industrial production process, where the aim is to optimally allocate the jobs in each machine. In its multiobjective version, there are formulations about the minimization of the amount of time required to process all jobs (makespan), the total flowtime of the jobs, among others objectives. There are several heuristics and metaheuristics available to solve Multi-objective Optimization Problems, however most of them require deep knowledge in order to implement them efficiently. The Hyper-Heuristics (HHs) emerged as a high-level approach to automatically select or generate heuristics to solve complex problems. This paper proposes an implementation of an operator selection HH in the MOEA/D algorithm to solve the multiobjective Flowshop problem. The results obtained considering Taillard's benchmark instances are promising.*

Resumo. *O Flowshop é um problema de sequenciamento presente no gerenciamento do processo de produção industrial, onde se busca alocar da melhor maneira as atividades em cada máquina. Na sua versão multiobjetivo existem formulações que modelam a minimização do tempo para a execução de todas as atividades (makespan), o tempo total de fluxo das atividades (total flowtime), entre outros objetivos. Atualmente existem diversas heurísticas e metaheurísticas disponíveis para a solução de Problemas de Otimização Multiobjetivo, porém a maior parte delas requer profundos conhecimentos para serem implementadas de maneira eficiente. As Hiper-Heurísticas (HHs) surgiram como uma metodologia de alto nível para a seleção ou geração automática de heurísticas para a solução de problemas complexos. Este trabalho propõe a implementação de uma HH para a seleção de operadores no algoritmo MOEA/D para o tratamento do Flowshop multiobjetivo. Os resultados obtidos considerando instâncias do benchmark de Taillard são promissores.*

1. Introdução

Diversos problemas do mundo real possuem objetivos conflitantes, como no processo de produção na indústria onde deve-se minimizar o tempo total das atividades, maximizando a eficiência da produção e mantendo a qualidade da produção. Esses problemas, contendo mais de uma função objetivo a serem minimizadas ou maximizadas, são denominados Problemas de Otimização Multiobjetivo (POMs) [Zhou et al. 2011]. A solução para esses

problemas não é única e, geralmente, é encontrado um conjunto de soluções compromisso entre os diferentes objetivos, chamado de Fronteira de Pareto [Coello et al. 2007].

Nem sempre é possível resolver POMs complexos de maneira exata. Nesse contexto, Algoritmos Evolucionários Multiobjetivo (AEMOs) tornam-se uma opção interessante. Técnicas de decomposição dos objetivos em AEMOs apresentam bons resultados na literatura, popularizando o uso de tais técnicas [Zhang et al. 2009]. O MOEA/D (*Multi-Objective Evolutionary Algorithm based on Decomposition*) é um dos mais relevantes dessa classe de algoritmos [Li et al. 2014]. No MOEA/D o problema é decomposto em problemas de otimização menores (mono-objetivo), e cada um deles é tratado por um agente de maneira colaborativa [Zhang e Li 2007, Trivedi et al. 2017].

O *Flowshop* é um problema de alocação de atividades na cadeia de produção de indústrias, onde se deseja programar n atividades a serem executadas por m máquinas diferentes [Yenisey e Yagmahan 2014]. Tratar a versão multiobjetivo do *Flowshop* é importante pois em sua maioria, os problemas dessa natureza (aplicados à indústria) são multiobjetivo e ainda existem muitas questões em aberto nesta área [Parveen e Ullah 2011]. Assim, minimizar o tempo de execução de todas as atividades (*makespan*), e o tempo total de uma tarefa (*flowtime*) diminui o custo de produção. Enquanto minimizar o atraso e maximizar o adiantamento de atividades, busca suprir melhor a demanda dos produtos [Yenisey e Yagmahan 2014].

Na literatura existem diversas heurísticas e metaheurísticas com bom desempenho na solução de diferentes POMs [Zhou et al. 2011]. Porém, muito conhecimento sobre o problema e sobre as técnicas é necessário para utilizá-las com eficiência. Em razão disso, surgiram as Hiper-Heurísticas, que diminuem a quantidade de conhecimento requerido para resolver problemas complexos de otimização [Burke et al. 2013]. Hiper-Heurísticas são técnicas que se propõem a serem suficientemente genéricas para serem aplicadas em diversos domínios diferentes. As alterações necessárias, geralmente, se limitam àquelas necessárias para atender aos novos domínios. As Hiper-heurísticas podem ser classificadas como sendo de seleção ou geração. As Hiper-Heurísticas de seleção visam escolher a melhor heurística a ser aplicada em cada instante do processo de otimização, enquanto as Hiper-Heurísticas de geração visam criar melhores heurísticas para resolverem o problema em questão. As Hiper-Heurísticas de seleção são capazes de selecionar (por meio de um heurística de alto nível) dentre uma gama de heurísticas (heurísticas de baixo nível), a que mais se adequa a um determinado problema [Burke et al. 2013].

Hiper-Heurísticas para problemas multiobjetivo, principalmente para problemas de otimização combinatoriais, ainda são pouco exploradas e apresentam resultados promissores na literatura [Burke et al. 2013]. Portanto, neste trabalho busca-se analisar o desempenho de uma Hiper-heurística de seleção para o *Flowshop*. Serão utilizados como heurísticas de baixo nível os operadores de cruzamento dos Algoritmos Genéticos (AGs), onde a Hiper-Heurística selecionará a cada passo, qual operador é o mais adequado a ser aplicado.

O restante do trabalho está organizado conforme a seguir. A Seção 2 descreve brevemente os conceitos relacionados a este trabalho. A descrição da Hiper-Heurística implementada é feita na Seção 3. A Seção 4 contém as simulações e os resultados obtidos. As conclusões e apontamentos para trabalhos futuros são apresentados na Seção 5.

2. Fundamentação Teórica

2.1. Problema de Otimização Multiobjetivo

Os Problemas de Otimização Multiobjetivo (POMs) apresentam mais de uma função objetivo que devem ser minimizadas ou maximizadas ao mesmo tempo [Zhou et al. 2011]. Tais funções objetivo comumente apresentam conflitos entre si. A solução para esses problemas não é única, e, geralmente, se encontra um conjunto de soluções compromisso entre os diferentes objetivos, chamado de Fronteira de Pareto [Zhou et al. 2011]. A definição de um POM é dada pela Equação 1 [Coello et al. 2007]:

$$\begin{aligned}
 & \text{Minimizar/Maximizar} : f(x) = (f_1(x), \dots, f_k(x)) \\
 & \text{sujeito a} : g_i(x) \leq 0, i = \{1, \dots, G\}, \text{ e} \\
 & \quad h_j(x) = 0, j = \{1, \dots, H\} \quad x \in \Omega \\
 & \quad x_i^{(L)} \leq x_i \leq x_i^{(U)} \quad i = 1, 2, \dots, n
 \end{aligned} \tag{1}$$

tal que $f(x)$ é um vetor contendo as funções objetivo do problema, sujeito a G e H que são as restrições de desigualdade e igualdade, respectivamente. A variável x é uma solução, e Ω o conjunto finito das soluções factíveis, $x_i^{(L)}$ o limite inferior e $x_i^{(U)}$ o limite superior das variáveis de decisão.

2.2. Flowshop

O *Flowshop* é um problema de alocação de atividades na cadeia de produção de indústrias, onde se deseja programar um conjunto de n atividades a serem executadas por um conjunto de m máquinas, e todas as n atividades devem passar pelas máquinas na mesma ordem, ou seja, a primeira atividade deve ser processada primeiramente na máquina 1, em seguida na 2, e assim sucessivamente até a máquina m . Assumindo: i) que cada tarefa somente é executada por uma a cada instante; ii) cada máquina processa uma tarefa por vez; iii) uma tarefa não pode ser interrompida; iv) todas as atividades são independentes; v) as máquinas estão sempre disponíveis e; vi) o tempo de preparação está incluso no tempo que leva para processar as atividades e não dependem da sequência de realização das atividades [Yenisey e Yagmahan 2014].

O objetivo desse problema é encontrar uma sequência de atividades, de modo a otimizar em algum aspecto a produção (função objetivo). Esse problema é definido como o *Flowshop* de Permutação, sendo um dos mais pesquisados no campo da pesquisa operacional [Minella et al. 2008, Fernandez-Viagas et al. 2017]. Neste trabalho são considerados os dois objetivos mais explorados na literatura e importantes na prática (por reduzirem o custo de produção): o *makespan* e *flowtime*.

O *makespan* é o objetivo mais estudado na literatura, denotado como $F/prmu/C_{max}$ e para instâncias com mais de duas máquinas pertence à classe de complexidade *NP-Difícil* [Minella et al. 2008]. Sendo definido por $C_{max}(\pi) = \max C_i(\pi), i = 1, \dots, n$, tal que C_i é o tempo de conclusão da atividade i em uma permutação π e n é o número total de atividades [Yenisey e Yagmahan 2014].

O segundo objetivo mais estudado é o *flowtime*, também sendo *NP-Difícil* para instâncias com mais de duas máquinas [Minella et al. 2008], denotado por

$F/\text{prmu}/\sum C_j$ e é definido por $F(\pi) = \sum_{i=1}^n (C_i(\pi))$, tal que F é o *flowtime* total das atividades em uma permutação π , C_i é o tempo de conclusão da i -ésima atividade e n é o total de atividades a serem realizadas [Yenisey e Yagmahan 2014].

Diversos algoritmos foram testados para o *Flowshop* e alguns deles são brevemente descritos a seguir. Uma busca tabu [Armentano e Claudio 2004] se mostrou efetiva e ressaltou que o uso da dominância de Pareto é mais eficiente que a aplicação de funções lineares. Um AG com busca local [Arroyo e Armentano 2005], apresentou um número maior de soluções com qualidade superior, quando comparado a algoritmos da literatura para instâncias de tamanhos maiores. Arroyo e de Souza Pereira (2011) desenvolveram uma heurística GRASP combinando-a com busca local, os resultados obtidos para várias instâncias provaram ser eficientes e competitivos na comparação com a literatura [Arroyo e de Souza Pereira 2011]. A utilização de modelos *Mallows* em Algoritmos de Estimativa de Probabilidade [Zangari et al. 2017], apresentou resultados competitivos com a literatura, com custo computacional e número de avaliações razoáveis para instâncias de tamanhos maiores. Os resultados obtidos para o *Flowshop*, aplicado a uma combinação de um AEMO com busca local, apontam para a importância da aplicação desta técnica devido ao aumento no desempenho dos algoritmos [Ishibuchi et al. 2003]. Um estudo aprofundado foi realizado sobre a utilização de vizinhos neutros para a exploração de soluções [Kessaci-Marmion et al. 2017], técnica que tem se mostrado eficiente para resolver POMs.

2.3. MOEA/D

O MOEA/D (*Multi-Objective Evolutionary Algorithm based on Decomposition*) é um algoritmo multiobjetivo baseado em decomposição. Ele decompõe um POM em subproblemas mono-objetivo. Os subproblemas são resolvidos simultaneamente em somente uma execução. A cada geração a população é formada pelas melhores soluções encontradas para cada subproblema até o momento. Cada subproblema possui um vetor de pesos de agregação, e a vizinhança é estabelecida pela distância euclidiana entre os vetores de cada subproblema. A melhor solução para vizinhos tende a ser similar e o MOEA/D explora essa característica solucionando cada subproblema com informações de sua vizinhança [Zhang e Li 2007].

Existem diversas maneiras de se decompor os objetivos [Trivedi et al. 2017]. Os métodos mais utilizados são: o de Tchebycheff, o PBI (*Penalty-based Boundary Intersection*) e a soma ponderada. A abordagem de Tchebycheff é fortemente empregada e é definida por Minimizar: $g^{te}(x|\lambda, z^*) = \max\{\lambda_i | f_i(x) - z_i^* | \}$, $1 \leq i \leq T$, sujeito a $x \in \Omega$ [Trivedi et al. 2017], tal que $z = (z_1, \dots, z_T)^T$ é o ponto de referência, $f_i(x)$ é a função multiobjetivo do problema. Cada ponto ótimo possui um vetor de pesos λ tal que x^* é uma solução ótima da função de agregação (decomposição) e cada solução x^* é também uma solução ótima de Pareto.

O algoritmo MOEA/D trata todos os subproblemas gerados da mesma maneira, alocando a mesma quantidade de recursos para resolvê-los. Porém, tais subproblemas podem apresentar diferentes complexidades e assim requerem uma quantidade diferente de recursos. O MOEA/D *with Dynamical Resource Allocation* (MOEA/D-DRA) [Zhang et al. 2009] foi criado para lidar com esse problema. Ele define e computa um valor utilitário π^i para cada subproblema i , desta forma os esforços computacionais são

atribuídos aos subproblemas com base em seus valores utilitários, quanto maior o valor utilitário associado a um subproblema maior a probabilidade dele ser escolhido.

2.4. Hiper-Heurísticas

Hiper-Heurísticas (HHs) são algoritmos de alto nível capazes de resolver diferentes instâncias de diferentes problemas sem modificações ou com modificações mínimas na implementação do algoritmo. As Hiper-Heurísticas podem ser classificadas como sendo de seleção ou geração. As Hiper-Heurísticas de seleção visam escolher a melhor heurística a ser aplicada em cada instante do processo de otimização, enquanto as Hiper-Heurísticas de geração visam criar melhores heurísticas para resolverem o problema em questão. A ênfase deste trabalho foi em Hiper-Heurísticas de seleção, visto que as mesmas tendem a apresentar melhores resultados práticos [Burke et al. 2013].

De acordo com o tipo de aprendizagem as HHs podem ser classificadas em *on-line* ou *off-line*. No aprendizado *on-line* os dados são obtidos durante a execução do algoritmo, enquanto que no aprendizado *off-line* estes vêm antes da execução do problema (*a priori*). A vantagem do método *on-line* é permitir que a Hiper-Heurística se adapte dinamicamente às condições correntes do processo de otimização. Além disso, o método *off-line* necessita que instâncias “típicas” dos problemas estejam disponíveis para o aprendizado, ou seja, se houver grande diferença entre as características das instâncias utilizadas no aprendizado e aquelas sendo resolvidas o desempenho da Hiper-Heurística tenderá a ser insatisfatório. Portanto, neste trabalho, optou-se pelo desenvolvimento de uma Hiper-Heurística com aprendizado *on-line*.

As HHs também devem decidir quando uma nova solução gerada deve ou não ser aceita. Esse componente das Hiper-Heurísticas é denominado política ou heurística de aceitação [Bilgin et al. 2006]. Neste trabalho, a heurística de aceitação corresponde ao método de seleção do MOEA/D.

2.5. Seleção Adaptativa de Operadores

A Seleção Adaptativa de Operadores (SAO) é um paradigma que surgiu da necessidade de escolher operadores adequados durante a geração de novas soluções em algoritmos evolucionários. Realizar essa seleção por tentativa e erro (método mais comumente empregado) tende a ser muito custoso computacionalmente, além de ineficiente quando existem variações destes operadores durante a execução do algoritmo [Venske et al. 2014]. Desta maneira o procedimento de SAO visa determinar qual o operador ideal para cada estágio do processo de busca, de acordo com o desempenho destes operadores [Li et al. 2014, Gonçalves et al. 2015].

Dois procedimentos principais fazem parte do SAO, o *credit assignment* e a seleção de operadores. O *credit assignment* define como recompensar determinado operador de acordo com seu desempenho recente.

A seleção de operadores é o processo de escolha de operadores que serão utilizados para gerar novas soluções de acordo com as recompensas atribuídas durante o *credit assignment*. Neste trabalho será utilizada uma abordagem baseada em *Multi-Armed Bandit* (MAB) [Li et al. 2014]. O problema do MAB, consiste em uma máquina com um determinado número de alavancas em que o apostador tem que escolher uma alavanca para puxar, de modo a maximizar seus ganhos em uma série de tentativas. Não existe

informação *a priori* sobre as alavancas e por meio de várias tentativas é que pode-se determinar quais alavancas são vantajosas [Robbins 1985].

No contexto de Hiper-Heurísticas pode-se considerar que cada heurística de baixo nível (operador) é uma alavanca cujo retorno (melhoria na qualidade das soluções) é inicialmente desconhecido. O objetivo da HH é escolher a heurística de melhor retorno e ao mesmo tempo obter informações suficientes sobre as demais heurísticas para se ter um nível de confiança razoável sobre a qualidade de todas as heurísticas.

Existem diversos algoritmos para resolver o MAB [Auer et al. 2002, Li et al. 2014]. Nesse trabalho o UCB1 foi adotado. Esse algoritmo utiliza o conceito de Upper Confidence Bound (UCB), ou seja, o algoritmo estima com alta probabilidade o limite superior para a recompensa de cada alavanca e escolhe a alavanca com maior limite superior. A exploração acontece para aumentar a confiabilidade dos limites estimados. O UCB1 é um algoritmo simples e com desempenho assintótico ótimo [Li et al. 2014]. Ele Cada alavanca (operador), deve ser utilizada pelo menos uma vez para que o algoritmo UCB1 seja aplicado. A escolha da alavanca é realizada de acordo com aquela que maximiza a Equação 2 [Auer et al. 2002]:

$$FRR + C \times \sqrt{\frac{2 \times \ln \sum_{j=1}^K n_{j,t}}{n_{i,t}}} \quad (2)$$

tal que FRR é uma estimativa de qualidade empírica (ver subseção 3.1), i a alavanca, t o tempo, C um fator para balancear a intensificação e a exploração do processo de aprendizado, os termos na raiz quadrada representam as incertezas associadas a cada alavanca. O número de vezes que a alavanca i foi utilizada é denotado por n e K o conjunto de alavancas.

3. Hiper-Heurística para o Flowshop - MOEA/D-DRA-MAB

Esta seção descreve a implementação da Hiper-Heurística de seleção baseada em *Multi-Armed Bandit* para o problema do *Flowshop* de Permutação. A adaptação dos elementos do MOEA/D para a solução do problema do *Flowshop* de Permutação foi a seguinte. Uma solução para problema é representada como uma permutação das atividades a serem realizadas. O Algoritmo 1 apresenta o pseudocódigo da HH implementada, nomeada MOEA/D-DRA-MAB, utilizando como base o algoritmo MOEA/D. Os primeiros cinco passos do algoritmo correspondem aos procedimentos de inicialização, nos quais os vetores de pesos, as vizinhanças, as soluções da população (geradas aleatoriamente) e os vetores especiais z e *nadir* são inicializados. Os vetores de pesos utilizados para cada subproblema foram uniformemente distribuídos e gerados conforme a técnica descrita em [Zhang et al. 2009].

Após essa fase o algoritmo entra no laço principal (passos 6 a 32), neste laço é feita a seleção de N_e (escopo) para cada cromossomo. No passo 12 a escolha do operador mais adequado é feita pela HH e no próximo passo uma nova solução é gerada utilizando este operador. Em seguida a mutação Swap é aplicada à nova solução. Os dois laços seguintes (passos 14 a 21) atualizam os pontos especiais z e *nadir*. O último laço (passos 23 a 30) atualiza a população. O critério de parada utilizado foi um número máximo de avaliações

da função objetivo. Considerando que o *Flowshop* é um problema de permutação, é importante a utilização de operadores adequados a essa classe de problemas. Desta forma foram utilizados os seguintes operadores de cruzamento [Larrañaga et al. 1999]: *Cycle Crossover* (CX), *Order Crossover* (OX), *Partially-Mapped Crossover* (PMX), e o operador de mutação *Swap* (troca).

3.1. Descrição da Hiper-Heurística Implementada

Os principais componentes de uma Hiper-Heurística, conforme a seção 2.4 são: mecanismo de seleção, critério de aceitação, critério de parada e as heurísticas de baixo nível

Algoritmo 1 Pseudocódigo do MOEA/D-DRA-MAB

```

1: Gerar os vetores de pesos  $\lambda^i = (\lambda_1^i, \lambda_2^i, \dots, \lambda_T^i), i = 1, \dots, Sp$ 
2: Calcular a distância Euclidiana entre os pares de vetores de pesos e encontrar os
   C vetores mais próximos  $(\lambda^i, \dots, \lambda^{i_C})$  para cada vetor de pesos  $\lambda^i$ . Para  $i =$ 
    $1, \dots, Sp, B(i) = \{i_1, \dots, i_C\}$ 
3: Gerar a população inicial  $\mathbf{x}^1, \dots, \mathbf{x}^{Sp}$ , aleatoriamente.  $\mathbf{VF}^i = F(\mathbf{x}^i)$ . Cada  $\mathbf{x}^i$  e  $\mathbf{VF}^i$  é
   associado a um vetor de pesos  $\lambda^i$ 
4: Inicializar z:  $z_j = \min_{objetivo_j}(\mathbf{VF}^i), i = 1, \dots, Sp, j = 1, \dots, T$ 
5: Inicializar nadir:  $nadir_j = \max_{objetivo_j}(\mathbf{VF}^i), i = 1, \dots, Sp, j = 1, \dots, T$ 
6: repeat
7:   for cada  $x^i$  do
8:     Gerar rand em U[0,1]
9:     if rand <  $\delta$  then  $Ne = B(i)$ 
10:    else  $Ne = \{1, \dots, Sp\}$ 
11:    end if
12:    Selecionar o Operador utilizando a HH
13:    Gerar nova solução  $\mathbf{x}'$  por meio do operador selecionado
14:    for cada  $j = 1, \dots, T$  do
15:      if  $z_j > F_j(\mathbf{x}'')$  then  $z_j = F_j(\mathbf{x}'')$ 
16:      end if
17:    end for
18:    for cada  $j = 1, \dots, T$  do
19:      if  $nadir_j < F_j(\mathbf{x}'')$  then  $nadir_j = F_j(\mathbf{x}'')$ 
20:      end if
21:    end for
22:    contador = 0
23:    while contador < maxAtualizações e  $\exists$  índices não selecionados de  $Ne$  do
24:      selecionar aleatoriamente um índice  $k$  de  $Ne$ 
25:      if  $g^{te}(\mathbf{x}'' | \lambda^k, z, nadir) \leq g^{te}(\mathbf{x}^k | \lambda^k, z, nadir)$  then
26:         $\mathbf{x}^k = \mathbf{x}''$ 
27:         $\mathbf{VF}^k = F(\mathbf{x}'')$ 
28:        contador = contador + 1
29:      end if
30:    end while
31:  end for
32: until Critério de parada satisfeito

```

implementadas (operadores genéticos). Para a implementação do mecanismo de seleção foi considerado o método de *Fitness Improvement Rate* (FIR), definido pela Equação 3 [Gonçalves et al. 2015]:

$$FIR_{op,t} = \frac{g^{te}(x^i|\lambda^i, z^*) - g^{te}(y|\lambda^i, z^*)}{g^{te}(x^i|\lambda^i, z^*)} \quad (3)$$

tal que $FIR_{op,t}$ é o valor alcançado pelo operador op no tempo t , g^{te} o *fitness* utilizando Tchebycheff, com x^i sendo o pai, y o filho, λ o vetor de pesos de i e z^* o ponto de referência. Para resolver o problema da variação de melhoria no *fitness* durante o processo de busca, uma janela deslizante implementada como fila é utilizada para guardar os operadores aplicados recentemente. A fim de aplicar o crédito ao devido operador, uma soma de todos os valores de FIR contidos na janela deslizante é efetuada, então um fator de decaimento é aplicado para que os melhores operadores tenham maiores chances, por fim esse valor é normalizado, gerando o *Fitness-Rate-Ranking* (FRR) que é então utilizado pela seleção de operadores, por intermédio do UCB1 (Equação 2).

Os critérios de aceitação e de parada foram implementados no MOEA/D descrito na seção 2.3. Uma solução é aceita caso ela seja selecionada pelo mecanismo de seleção do MOEA/D para compor a próxima população. O critério de parada é verificado pelo número de avaliações estabelecido. Os operadores implementados foram o PMX, OX e CX. Após a aplicação do operador selecionado pela HH, a mutação *swap* é realizada.

4. Simulações e Resultados

Esta seção contém os resultados obtidos com 30 execuções dos algoritmos considerados utilizando 300.000 avaliações como critério de parada. A Subseção 4.1 apresenta o ajuste de parâmetros. A Subseção 4.2 compara a qualidade dos resultados da Hiper-Heurística proposta (MOEA/D-DRA-MAB) com relação aos operadores aplicados isoladamente. Em todas as tabelas os valores em cinza escuro significam os melhores resultados e os valores em cinza claro correspondem ao segundo melhor resultado. Para a execução das simulações foram utilizadas 12 instâncias do *benchmark* de Taillard, contendo de 20 a 500 atividades e de 5 a 20 máquinas [Taillard 1993]. Para medir o desempenho dos algoritmos são utilizados dois indicadores de qualidade o Hipervolume e ε -Unário. O Hipervolume mede o volume do espaço de busca que é dominado pela aproximação da fronteira de Pareto enquanto o ε -Unário mede o quanto é necessário deslocar uma aproximação da fronteira para dominar uma fronteira de referência. O primeiro indicador deve ser maximizado enquanto o segundo deve ser minimizado [Zitzler et al. 2008].

4.1. Ajuste de Parâmetros

Esta seção apresenta testes para o ajuste de alguns parâmetros do algoritmo proposto considerando o indicador Hipervolume. A Tabela 1 exibe os resultados do indicador Hipervolume para diferentes tamanhos de população. A configuração com 50 indivíduos obteve os melhores resultados para quatro instâncias, enquanto a configuração com 100 indivíduos obteve os melhores resultados para cinco instâncias. Três instâncias obtiveram resultado melhor com a população de 300 soluções. Portanto, todas as simulações subsequentes foram realizadas considerando o tamanho da população igual a 100 soluções.

Tabela 1. Hipervolume - Média e desvio padrão - tamanhos de População.

	50 soluções	100 soluções	300 soluções
tai20_5.txt	$2.00e - 03_{1.1e-02}$	$1.37e - 02_{4.2e-02}$	$1.54e - 02_{5.0e-02}$
tai20_10.txt	$6.36e - 02_{1.2e-01}$	$5.40e - 02_{9.6e-02}$	$7.19e - 02_{1.0e-01}$
tai20_20.txt	$1.44e - 01_{1.1e-01}$	$1.49e - 01_{1.2e-01}$	$1.29e - 01_{1.1e-01}$
tai50_5.txt	$1.38e - 02_{4.2e-02}$	$3.34e - 02_{7.9e-02}$	$2.85e - 02_{5.4e-02}$
tai50_10.txt	$5.57e - 02_{1.2e-01}$	$5.15e - 02_{1.2e-01}$	$5.95e - 02_{1.1e-01}$
tai50_20.txt	$7.14e - 02_{1.2e-01}$	$9.53e - 02_{1.8e-01}$	$7.33e - 02_{1.2e-01}$
tai100_5.txt	$1.10e - 01_{1.5e-01}$	$1.15e - 01_{1.1e-01}$	$7.86e - 02_{1.0e-01}$
tai100_10.txt	$5.22e - 02_{1.5e-01}$	$3.68e - 02_{7.0e-02}$	$7.14e - 03_{2.5e-02}$
tai100_20.txt	$1.27e - 01_{1.3e-01}$	$1.29e - 01_{1.3e-01}$	$1.28e - 01_{1.5e-01}$
tai200_10.txt	$2.71e - 01_{1.6e-01}$	$2.15e - 01_{1.7e-01}$	$9.54e - 02_{1.0e-01}$
tai200_20.txt	$6.89e - 02_{1.5e-01}$	$4.80e - 02_{1.1e-01}$	$4.66e - 04_{2.5e-03}$
tai500_20.txt	$5.82e - 02_{1.1e-01}$	$1.45e - 02_{5.3e-02}$	$1.94e - 03_{1.0e-02}$

A Tabela 2 apresenta os resultados do indicador Hipervolume para diferentes taxas de mutação. A tabela mostra que para 11 instâncias este indicador apresenta o melhor resultado para a taxa de 50% de mutação, a qual foi adotada no restante das simulações. A

Tabela 2. Hipervolume - Média e desvio padrão - taxas de mutação.

	10% mutação	30% mutação	50% mutação
tai20_5	$0.00e + 00_{0.0e+00}$	$1.16e - 02_{5.3e-02}$	$1.92e - 02_{5.3e-02}$
tai20_10	$1.07e - 01_{1.1e-01}$	$1.88e - 01_{1.4e-01}$	$1.94e - 01_{1.5e-01}$
tai20_20	$5.55e - 02_{9.0e-02}$	$7.50e - 02_{1.0e-01}$	$1.00e - 01_{1.1e-01}$
tai50_5	$3.21e - 02_{8.1e-02}$	$5.14e - 02_{1.3e-01}$	$3.31e - 02_{1.1e-01}$
tai50_10	$3.00e - 02_{5.4e-02}$	$6.44e - 02_{1.2e-01}$	$1.03e - 01_{1.1e-01}$
tai50_20	$2.22e - 01_{1.1e-01}$	$2.90e - 01_{1.7e-01}$	$3.26e - 01_{1.4e-01}$
tai100_5	$2.39e - 01_{1.8e-01}$	$2.82e - 01_{1.7e-01}$	$3.98e - 01_{1.9e-01}$
tai100_10	$9.10e - 02_{1.4e-01}$	$1.37e - 01_{1.9e-01}$	$1.82e - 01_{2.0e-01}$
tai100_20	$9.65e - 02_{8.8e-02}$	$2.63e - 01_{1.5e-01}$	$3.22e - 01_{1.6e-01}$
tai200_10	$1.78e - 06_{9.6e-06}$	$2.16e - 02_{3.8e-02}$	$8.39e - 02_{1.0e-01}$
tai200_20	$4.16e - 03_{1.8e-02}$	$7.32e - 02_{1.4e-01}$	$1.36e - 01_{1.5e-01}$
tai500_20	$0.00e + 00_{0.0e+00}$	$2.08e - 03_{1.1e-02}$	$2.86e - 02_{1.3e-01}$

Tabela 3 apresenta os resultados, considerando diferentes taxas de cruzamento. A taxa de cruzamento de 100% obteve o melhor resultado para sete instâncias e o segundo melhor para três instâncias. Desta maneira, a taxa de cruzamento foi fixada em 100%. Assim, todos os testes subsequentes utilizam os seguintes parâmetros: população com 100 indivíduos e com 50% e 100% a taxa de mutação e cruzamento, respectivamente.

Tabela 3. Hipervolume - Média e desvio padrão - taxas de cruzamento.

	50% cruzamento	75% cruzamento	100% cruzamento
tai20_5	$5.10e - 03_{2.7e-02}$	$8.46e - 03_{4.6e-02}$	$8.60e - 03_{4.4e-02}$
tai20_10	$1.64e - 01_{1.3e-01}$	$2.10e - 01_{1.1e-01}$	$1.74e - 01_{1.4e-01}$
tai20_20	$1.65e - 01_{1.3e-01}$	$2.07e - 01_{1.4e-01}$	$2.34e - 01_{1.1e-01}$
tai50_5	$1.08e - 01_{1.9e-01}$	$1.20e - 01_{1.8e-01}$	$8.17e - 02_{1.6e-01}$
tai50_10	$1.36e - 02_{3.2e-02}$	$1.97e - 02_{8.0e-02}$	$2.28e - 02_{7.0e-02}$
tai50_20	$3.79e - 01_{2.2e-01}$	$3.53e - 01_{2.0e-01}$	$3.65e - 01_{2.1e-01}$
tai100_5	$5.50e - 02_{8.7e-02}$	$6.84e - 02_{1.3e-01}$	$4.80e - 02_{1.0e-01}$
tai100_10	$1.46e - 01_{1.9e-01}$	$1.17e - 01_{1.6e-01}$	$1.48e - 01_{2.1e-01}$
tai100_20	$1.19e - 01_{1.6e-01}$	$5.69e - 02_{1.1e-01}$	$9.67e - 02_{1.8e-01}$
tai200_10	$2.19e - 01_{1.3e-01}$	$2.13e - 01_{1.7e-01}$	$2.74e - 01_{1.9e-01}$
tai200_20	$2.05e - 02_{6.4e-02}$	$1.42e - 02_{5.6e-02}$	$2.51e - 02_{7.7e-02}$
tai500_20	$8.00e - 02_{1.4e-01}$	$7.26e - 02_{1.5e-01}$	$1.12e - 01_{1.4e-01}$

4.2. Efeito da Hiper-Heurística

Esta seção apresenta os testes realizados a fim de identificar o efeito da seleção adaptativa realizada pelo MOEA/D-DRA-MAB, com relação ao algoritmo MOEA/D-DRA utilizando os operadores (PMX, OX e CX) separadamente. As Tabelas 4 e 5 contêm a média e

o desvio padrão para os indicadores ε -Unário e Hipervolume, respectivamente. A Tabela 4 apresenta os resultados do indicador ε -Unário. Na tabela o valores em cinza escuro representam os melhores resultados e os valores em cinza claro indicam equivalência estatística de acordo com o teste Mann-Whitney-Wilcoxon com 95% de confiança [Conover 1999]. O algoritmo proposto, MOEA/D-DRA-MAB obtém os melhores resultados para 6 das 12 instâncias é equivalente estatisticamente ao melhor algoritmo em 5 das demais instâncias.

Tabela 4. ε -Unário - Média e desvio padrão - efeito da Hiper-Heurística.

	MOEA/D-DRA-MAB	MOEA/D-DRA-PMX	MOEA/D-DRA-OX	MOEA/D-DRA-CX
tai20_5.txt	1.66e + 02 _{1.2e+02}	1.68e + 02 _{1.0e+02}	1.99e + 02 _{1.2e+02}	2.14e + 02 _{9.9e+01}
tai20_10.txt	3.76e + 02 _{2.8e+02}	3.20e + 02 _{1.8e+02}	3.11e + 02 _{2.1e+02}	4.96e + 02 _{2.2e+02}
tai20_20.txt	3.80e + 02 _{1.8e+02}	3.37e + 02 _{1.9e+02}	3.94e + 02 _{2.1e+02}	4.43e + 02 _{1.5e+02}
tai50_5.txt	1.13e + 03 _{5.8e+02}	1.04e + 03 _{5.7e+02}	9.64e + 02 _{4.4e+02}	9.08e + 02 _{4.2e+02}
tai50_10.txt	2.00e + 03 _{7.7e+02}	2.05e + 03 _{9.1e+02}	2.09e + 03 _{8.0e+02}	2.06e + 03 _{7.4e+02}
tai50_20.txt	1.75e + 03 _{9.6e+02}	2.73e + 03 _{1.1e+03}	2.14e + 03 _{9.5e+02}	2.35e + 03 _{1.1e+03}
tai100_5.txt	3.07e + 03 _{1.1e+03}	3.13e + 03 _{1.3e+03}	3.31e + 03 _{1.3e+03}	3.25e + 03 _{1.4e+03}
tai100_10.txt	4.91e + 03 _{2.3e+03}	3.82e + 03 _{2.1e+03}	5.45e + 03 _{2.0e+03}	4.28e + 03 _{1.6e+03}
tai100_20.txt	5.91e + 03 _{2.7e+03}	6.54e + 03 _{2.6e+03}	7.08e + 03 _{2.3e+03}	6.17e + 03 _{2.6e+03}
tai200_10.txt	1.02e + 04 _{2.4e+03}	1.07e + 04 _{4.4e+03}	1.38e + 04 _{5.3e+03}	1.35e + 04 _{4.9e+03}
tai200_20.txt	1.50e + 04 _{6.0e+03}	1.42e + 04 _{7.4e+03}	1.60e + 04 _{6.9e+03}	1.34e + 04 _{6.3e+03}
tai500_20.txt	4.25e + 04 _{1.8e+04}	4.00e + 04 _{2.0e+04}	8.32e + 04 _{2.6e+04}	7.36e + 04 _{2.7e+04}

A Tabela 5 apresenta os resultados do indicador Hipervolume seguindo a mesma metodologia adotada na Tabela 4. Essa tabela indica que a hiper-heurística proposta é equivalente ao melhor algoritmo para todas as instâncias com 50 e 100 trabalhos, contudo para instâncias com 200 trabalhos a versão MOEA/D-DRA-PMX mostrou-se superior. Como a versão MOEA/D-DRA-OX também mostrou bons resultados e o MOEA/D-DRA-OX foi a pior versão fica evidente que para esse problema a posição absoluta dos valores é mais importante do que a posição relativa dos mesmos.

Tabela 5. Hipervolume - Média e desvio padrão - efeito do MOEA/D-DRA-MAB.

	MOEA/D-DRA-MAB	MOEA/D-DRA-PMX	MOEA/D-DRA-OX	MOEA/D-DRA-CX
tai20_5.txt	8.97e - 03 _{4.6e-02}	6.57e - 03 _{3.5e-02}	8.10e - 03 _{4.4e-02}	4.76e - 04 _{2.6e-03}
tai20_10.txt	2.49e - 01 _{1.6e-01}	3.18e - 01 _{1.3e-01}	3.38e - 01 _{1.2e-01}	1.41e - 01 _{1.0e-01}
tai20_20.txt	1.99e - 01 _{1.4e-01}	1.23e - 01 _{1.1e-01}	1.83e - 01 _{1.4e-01}	6.84e - 02 _{9.3e-02}
tai50_5.txt	6.64e - 01 _{1.2e-01}	6.84e - 01 _{1.1e-01}	6.87e - 01 _{1.0e-01}	6.67e - 01 _{9.5e-02}
tai50_10.txt	2.18e - 01 _{2.2e-01}	2.35e - 01 _{1.9e-01}	2.09e - 01 _{2.0e-01}	1.87e - 01 _{2.1e-01}
tai50_20.txt	1.74e - 01 _{2.1e-01}	2.11e - 01 _{2.1e-01}	1.33e - 01 _{1.7e-01}	1.50e - 01 _{1.4e-01}
tai100_5.txt	4.50e - 01 _{1.3e-01}	4.63e - 01 _{1.1e-01}	4.08e - 01 _{1.4e-01}	4.01e - 01 _{1.3e-01}
tai100_10.txt	3.50e - 01 _{1.8e-01}	3.28e - 01 _{1.3e-01}	2.78e - 01 _{1.2e-01}	2.12e - 01 _{1.2e-01}
tai100_20.txt	1.93e - 01 _{1.4e-01}	1.74e - 01 _{1.4e-01}	1.28e - 01 _{1.4e-01}	1.78e - 01 _{1.3e-01}
tai200_10.txt	4.92e - 01 _{1.6e-01}	5.35e - 01 _{1.1e-01}	3.23e - 01 _{1.2e-01}	3.43e - 01 _{1.3e-01}
tai200_20.txt	1.70e - 02 _{6.7e-02}	3.24e - 02 _{1.4e-01}	6.42e - 03 _{3.1e-02}	7.63e - 03 _{3.1e-02}
tai500_20.txt	1.32e - 02 _{7.1e-02}	2.60e - 02 _{1.4e-01}	0.00e + 00 _{0.0e+00}	0.00e + 00 _{0.0e+00}

5. Conclusões e Trabalhos Futuros

Este trabalho apresentou o desenvolvimento e análise de uma Hiper-Heurística de seleção para o *Flowshop* de Permutação multiobjetivo. O algoritmo proposto, o MOEA/D-DRA-MAB, foi testado em 12 instâncias do *benchmark* de Taillard que consideram diferentes quantidades de máquinas e atividades. Os experimentos foram conduzidos em duas etapas e levaram em consideração dois indicadores de qualidade. Na primeira etapa os parâmetros do algoritmo MOEA/D foram empiricamente ajustados: tamanho da

população, taxa de cruzamento e taxa de mutação. A seguir, os operadores individuais foram confrontados com a escolha automática feita pela Hiper-Heurística. Neste conjunto de testes o algoritmo MOEA/D-DRA-MAB obteve desempenho satisfatório obtendo o melhor ou o segundo melhor resultado, com a vantagem de não exigir que o usuário escolha o operador que deve ser aplicado ao longo do processo de busca.

Os resultados são promissores e foram identificados vários pontos a serem explorados. A literatura mostra diferentes estratégias e componentes que podem melhorar o desempenho de AEMOs, tais como: o uso de buscas locais, o uso de um arquivo externo para o armazenamento das soluções não dominadas e diferentes funções de agregação.

Agradecimentos

Os autores agradecem ao CNPq e à Fundação Araucária pelo apoio financeiro.

Referências

- Armentano, V. A. e Claudio, J. E. (2004). An application of a multi-objective tabu search algorithm to a bicriteria flowshop problem. *Journal of Heuristics*, 10(5):463–481.
- Arroyo, J. E. C. e Armentano, V. A. (2005). Genetic local search for multi-objective flowshop scheduling problems. *European Journal of operational research*, 167(3):717–738.
- Arroyo, J. E. C. e de Souza Pereira, A. A. (2011). A grasp heuristic for the multi-objective permutation flowshop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 55(5):741–753.
- Auer, P., Cesa-Bianchi, N., e Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.
- Bilgin, B., Özcan, E., e Korkmaz, E. E. (2006). An experimental study on hyper-heuristics and exam timetabling. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 394–412. Springer.
- Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., e Qu, R. (2013). Hyper-heuristics: A survey of the state of the art. *Journal of the Operational Research Society*, 64(12):1695–1724.
- Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A., et al. (2007). *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer.
- Conover, W. J. (1999). *Practical Nonparametric Statistics*. Wiley, 3 edition.
- Fernandez-Viagas, V., Ruiz, R., e Framinan, J. M. (2017). A new vision of approximate methods for the permutation flowshop to minimise makespan: State-of-the-art and computational evaluation. *European J. of Operational Research*, 257(3):707–721.
- Gonçalves, R. A., Almeida, C. P., e Pozo, A. (2015). Upper confidence bound (ucb) algorithms for adaptive operator selection in moea/d. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 411–425. Springer.
- Ishibuchi, H., Yoshida, T., e Murata, T. (2003). Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE transactions on evolutionary computation*, 7(2):204–223.

- Kessaci-Marmion, M.-E., Dhaenens, C., e Humeau, J. (2017). Neutral neighbors in bi-objective optimization: Distribution of the most promising for permutation problems. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 344–358. Springer.
- Larrañaga, P., Kuijpers, C., Murga, R., Inza, I., e Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2):129–170.
- Li, K., Fialho, A., Kwong, S., e Zhang, Q. (2014). Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 18(1):114–130.
- Minella, G., Ruiz, R., e Ciavotta, M. (2008). A review and evaluation of multiobjective algorithms for the flowshop scheduling problem. *INFORMS Journal on Computing*, 20(3):451–471.
- Parveen, S. e Ullah, H. (2011). Review on job-shop and flow-shop scheduling using. *Journal of Mechanical Engineering*, 41(2):130–146.
- Robbins, H. (1985). Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer.
- Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64(2):278–285.
- Trivedi, A., Srinivasan, D., Sanyal, K., e Ghosh, A. (2017). A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3):440–462.
- Venske, S. M., Gonçalves, R. A., e Delgado, M. R. (2014). Ademo/d: Multiobjective optimization by an adaptive differential evolution algorithm. *Neurocomputing*, 127:65–77.
- Yenisey, M. M. e Yagmahan, B. (2014). Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. *Omega*, 45:119–135.
- Zangari, M., Mendiburu, A., Santana, R., e Pozo, A. (2017). Multiobjective decomposition-based mallows models estimation of distribution algorithm. a case of study for permutation flowshop scheduling problem. *Inform. Sciences*, 397:137–154.
- Zhang, Q. e Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.
- Zhang, Q., Liu, W., e Li, H. (2009). The performance of a new version of moea/d on cec09 unconstrained mop test instances. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 203–208. IEEE.
- Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., e Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49.
- Zitzler, E., Knowles, J., e Thiele, L. (2008). Quality assessment of pareto set approximations. *Multiobjective Optimization*, pages 373–404.

Um Estudo Experimental da Aplicação do Algoritmo de Neuroevolução com Crescimento Topológico em Imagens Médicas

Luiz D.R. França¹, Péricles B.C. Miranda¹, Filipe R. Cordeiro¹

¹ Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco (UFRPE) – Recife – PE – Brazil

{luizdaniel.r.f, periclesmiranda, filipe.rolim}@gmail.com

Abstract. *Breast cancer is one of the diseases that most affects women. Diagnosis of the disease in the early stages increases the chances of success in treatments. For this purpose several classification algorithms are studied and applied in this context. This work proposes an experimental study of the application of the algorithm of Neuroevolution of Augmenting Topologies (NEAT) to detect cancer in medical images. The proposal was evaluated for accuracy being applied to different medical imaging databases. The NEAT achieved promising results when compared to different classifiers widely used in the literature.*

Resumo. *O câncer de mama é uma das doenças que mais afeta as mulheres. O diagnóstico da doença nos estágios iniciais aumenta as chances de sucesso nos tratamentos. Com este objetivo vários algoritmos de classificação são estudados e aplicados neste contexto. Este trabalho propõe um estudo experimental da aplicação do algoritmo de Neuroevolução com Aumento Topológico (NEAT) para detecção de câncer em imagens médicas. A proposta foi avaliada quanto à precisão sendo aplicada a diferentes bases de imagens médicas. O NEAT alcançou resultados promissores quando comparado com diferentes classificadores bastante utilizados na literatura.*

1. Introdução

O câncer de mama ocorre quando as células da mama se reproduzem fora de controle. De acordo com a *American Cancer Society*¹, a doença pode começar nos canais que levam leite até o mamilo, chamado de Carcinoma Ductal Invasivo, ou nas glândulas que produzem leite, chamado de Carcinoma Lobular Invasivo. Nos estágios mais avançados, pode se espalhar pelo corpo em um processo chamado de metástase, podendo levar até a morte. De acordo com a *World Health Organization (WHO)*², câncer de mama é o tipo de câncer que mais mata mulheres de 20 à 59 anos no mundo e em 2011 foi a décima principal causa de morte entre mulheres.

É de suma importância que o câncer de mama seja diagnosticado o mais cedo possível. De acordo com um estudo feito pelo *Office for National Statistics*³ de 2014 à 2015, 95% das mulheres diagnosticadas com câncer de mama no estágio 3 sobreviveram

¹<https://www.cancer.org>

²<http://www.who.int>

³<https://www.ons.gov.uk/>

ao primeiro ano após terem sido diagnosticadas contra apenas 63% das mulheres diagnosticadas no estágio 4. E segundo a *American Cancer Society*, 100% das mulheres diagnosticadas no estágio 1 vivem por pelo menos cinco anos após terem sido diagnosticadas, enquanto apenas 22% das mulheres diagnosticadas no estágio 4.

Imagens médicas são muito utilizadas para auxiliar o especialista na detecção de câncer de mama em mulheres que ainda não apresentam sinais ou sintomas da doença. Dependendo da experiência do especialista, da qualidade da imagem e da idade da mulher, existe a possibilidade do exame gerar um falso-positivo, o que pode gerar ansiedade e estresse para o paciente, além da necessidade de outros exames que podem ser mais invasivos. Mais de 50% das mulheres que realizam exames anualmente vão receber falso-positivos, segundo o *National Cancer Institute*⁴.

Perante o exposto, este trabalho propõe um estudo experimental da eficiência do algoritmo de Neuroevolução com Aumento Topológico (NEAT) quando comparado com outros algoritmos de classificação para a detecção de câncer de mama em imagens médicas. Foi avaliada a precisão do NEAT comparando-o com resultados das redes neurais artificiais do tipo *Multilayer-perceptron* (MLP) e *Radial Basis Function* (RBF), e com os algoritmos *Random Forest* e *Support Vector Machine* (SVM). Foi utilizada a base de imagens médicas *Image Retrieval in Medical Applications* (IRMA) [Oliveira et al. 2008], com mais de 20.000 imagens, utilizando os seguintes descritores: *Gray Level Co-Occurrence Matrices* (GLCM), *Local Binary Patterns* (LBP) e *Zernike*. Também será usada a base de imagens dermatoscópicas *International Skin Imaging Collaboration* (ISIC) [Gutman et al. 2016] com 571 instâncias, utilizando os descritores GLCM e LBP. E por fim, a base de dados *Wisconsin Breast Cancer Diagnostic* (WDBC) [Wolberg et al. 1992] contendo 569 instâncias. Os resultados dos experimentos mostram que o NEAT pode ser uma boa alternativa para a classificação de tumores em imagens médicas.

O artigo está organizado como segue: na seção 2 é apresentada a contextualização do problema. A seção 3 explica o algoritmo NEAT. Na seção 4 a metodologia experimental é detalhada. Na seção 5 os resultados são apresentados. Por fim, conclusão e trabalhos futuros são discutidos na seção 6.

2. Trabalhos Relacionados

Dentre os algoritmos já utilizados para a classificação de tumores em imagens de mamografia, destacam-se as Redes Neurais Artificiais (RNAs) [Ibrahim et al. 2015]. Embora as RNAs tenham sido utilizadas com sucesso em diferentes problemas de classificação, o seu desempenho depende do adequado ajuste de seus hiper-parâmetros, tais como, número de camadas, pesos, entre outros. Deste modo, vários trabalhos passaram a utilizar abordagens inteligentes com o intuito de otimizar o desempenho dos algoritmos de classificação na detecção de câncer.

O trabalho desenvolvido por [Ibrahim et al. 2015] propõe a adoção de MLP com o algoritmo multiobjetivo de Evolução Diferencial (DE) para diagnóstico câncer de mama. Nesse artigo DE é utilizado para otimizar a precisão da MLP e a complexidade da topologia (e.g. quantidade de nós na camada escondida). Já o trabalho de Dheeba

⁴<https://www.cancer.gov>

et al. [Dheeba et al. 2014] aplica *Particle Swarm Optimized Wavelet Neural Network* (PSOWNN) para detecção de anormalidades em imagens de mamografia. O algoritmo PSO é usado para melhorar a precisão do WNN diminuindo a taxa de erro.

Zohra et al. [Zohra and Nacéra 2013] fazem uso de Algoritmos Genéticos de Múltiplas populações (MPGA) para otimizar redes neurais RBF para detecção de tumor em imagens de mamografia. O algoritmo *K-means* é usado para encontrar os centróides usados pela RBF e o MPGA é utilizado para evoluir os pesos da segunda camada da RBF. A função objetivo utilizada foi o erro quadrático.

No trabalho de Bhardwaj et al. [Bhardwaj and Tiwari 2015] é proposta uma Rede Neural Geneticamente Otimizada (GONN). Eles usam programação genética (PG) para criar redes neurais capazes de detectar câncer. A PG adotou como função objetivo o erro médio quadrático.

Os trabalhos de [Ahmad et al. 2015, Belciug and Gorunescu 2013] propuseram a aplicação de um algoritmo genético (GA) para a otimização da rede MLP quando aplicação à detecção de tumor. O primeiro foi utilizado para a seleção de características e criação da estrutura da MLP. O segundo aplicou o GA para otimizar os pesos da MLP com uma estrutura fixa.

Ahmad et al [Ahmad et al. 2012] adota *Cartesian Genetic Programming* para evoluir redes neurais artificiais (CGPANN) para detecção de câncer de mama. É usado um vetor de duas dimensões para representar os nós e conexões. O CGPANN considera não só os pesos como a topologia e escolhe a função de ativação mais apropriada. Em [Manning and Walsh 2013], é proposta uma melhoria para o CGPANN de Ahmad [Ahmad et al. 2012], introduzindo RBF e cruzamento (do inglês, *crossover*). Embora a proposta tenha apresentado uma alta precisão, ela apresenta uma limitação quanto ao limite máximo de nós, e máximo número de entradas recebidas por cada nó.

Além dos trabalhos apresentados anteriormente, vale destacar a pesquisa desenvolvida por Turabieh [Turabieh 2016]. Neste trabalho, o algoritmo NEAT foi comparado com a rede MLP na detecção de câncer. Embora o NEAT tenha atingido resultados melhores, o experimento é inconclusivo pois foi adotada apenas uma única base de dados para avaliação dos algoritmos.

Embora os trabalhos apresentados previamente tenham contribuído para a área, estes utilizaram, em seus experimentos, bases de dados pequenas (como a base de *Diagnostic Wisconsin Breast Cancer Database* e *Original Wisconsin Breast Cancer Database*). Além disso, alguns trabalhos supracitados impuseram limites quanto à otimização do classificador, por exemplo, número reduzido de parâmetros considerados, ou número fixo de nós na topologia da rede. Desta forma, se faz necessário um estudo mais profundo da otimização de algoritmos de detecção de tumores.

Diferentemente das pesquisas anteriores, este trabalho utiliza a base de dados *Image Retrieval in Medical Applications* (IRMA) [Oliveira et al. 2008], que apresenta mais de 20.000 imagens médicas, a base de imagens dermatoscópicas *International Skin Imaging Collaboration* (ISIC) [Gutman et al. 2016] com 571 instâncias e a base de dados *Wisconsin Breast Cancer Diagnostic* (WDBC) [Wolberg et al. 1992] contendo 569 instâncias; E não impõe limites quanto à otimização do algoritmo de detecção.

3. Neuroevolução com Aumento Topológico

Diferentes abordagens foram utilizadas para a otimização de RNAs no contexto de problemas de classificação, como exemplificados na seção anterior. Entre os trabalhos supracitados, vale destacar os algoritmos de neuroevolução [Stanley and Miikkulainen 2002]. Estes algoritmos vêm chamando atenção de pesquisadores por seu poder na otimização da topologia e hiper-parâmetros de RNAs.

Algoritmos de neuroevolução fazem uso de algoritmos genéticos para otimizar modelos de redes neurais, visando encontrar aquele mais adequado para um dado problema. É importante salientar que algoritmos de neuroevolução tradicionais buscam otimizar hiper-parâmetros da rede, tais como, definir as ligações e os pesos destas ligações entre neurônios de uma rede com uma topologia fixa. Porém, sabe-se que a topologia da rede também têm impacto na eficiência do algoritmo.

O NEAT é um algoritmo de neuroevolução aprimorado que constrói a topologia da RNA (ligações entre nós e pesos) de forma incremental. A evolução da rede neural parte de indivíduos simples, com apenas a camada de entrada ligada a camada de saída. Esses indivíduos são evoluídos sendo adicionados nós e ligações entre os nós. Para conseguir isto, o NEAT precisa resolver alguns problemas: Como representar a topologia de forma a permitir o cruzamento de topologias diferentes? Como proteger inovações topológicas que precisam de tempo para serem otimizadas de forma com que elas não sejam excluídas da população prematuramente? Como minimizar a estrutura da RNA durante a evolução sem a necessidade de colocar restrições ou métricas de complexidade na função objetivo?. O Algoritmo 1 apresenta as instruções do NEAT para a otimização da topologia de RNAs. Em seguida, cada etapa deste algoritmo será detalhada.

Algoritmo 1: Etapas do NEAT

Entrada: *max_it*, número máximo de gerações
Saída: população otimizada

```

1 início
2   pop = inicialização da população;
3   n_geracoes = 0;
4   repita
5     calcular fitness de pop;
6     especies = separar a população em especies;
7     para cada especie ∈ especies faça
8       selecionar os organismos pai;
9       realizar o crossover;
10      mutação dos organismos;
11      adicionar organismos a pop;
12      selecionar organismos para sobreviver de pop;
13    fim
14  até n_geracoes ≥ max_it;
15 fim

```

Para representar o genoma da RNA, o NEAT utiliza a codificação direta (e.g. todas as conexões e nós da rede são representadas no genoma) como mostra a Figura 1. O NEAT separa o genoma em duas listas, uma com a representação dos nós e outra com as

conexões. Os genes de nó contém a informação se o nó é um sensor (e.g. nó da camada de entrada), um nó da camada escondida ou um nó da camada de saída. Os genes de conexão representam qual o nó de entrada, qual o nó de saída, o peso da conexão, o número da inovação e se o gene está ativo ou não.

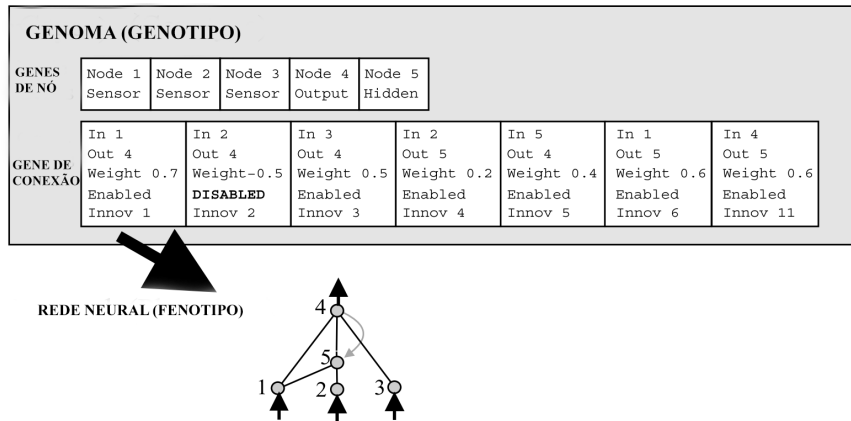


Figura 1. Exemplo de tradução do genoma para a rede neural retirada do artigo de [Stanley and Miikkulainen 2002]

A mutação pode ocorrer tanto para mutar os pesos das conexões como para adicionar novos nós ou conexões a topologia, como demonstrado na Figura 2. Na mutação para adicionar uma nova conexão, uma nova conexão é feita entre dois nós que não eram conectados com o peso designado aleatoriamente. Na mutação de adição de nós, uma conexão existente é dividida em dois. A conexão é desativada, e duas novas conexões são feitas, uma ligando o nó de entrada da conexão antiga ao novo nó, e o peso designado é de 1. A conexão entre o novo nó e o nó de saída da antiga conexão é criada utilizando o mesmo peso da conexão antiga. Desta forma, se evita de criar nós que não estão conectados a nenhum outro nó.

Para resolver o problema de representar uma estrutura de forma a permitir o cruzamento entre topologias diferentes, o NEAT utiliza marcadores históricos de inovação. Os marcadores históricos de inovação permitem identificar quais genes casam com qual, sendo assim dois genes com a mesma origem histórica deve representar a mesma conexão entre nós, podendo ter pesos diferentes. Para isso, toda vez que um novo gene, representando uma nova conexão, é criado um marcador de inovação é associado a ele. O NEAT faz uso do marcador de inovação para poder realizar o cruzamento entre organismos de topologias diferentes. O cruzamento é feito alinhando os genes que têm o mesmo número de inovações nos organismos pais. Os genes que têm o mesmo número de inovações são escolhidos aleatoriamente, enquanto os genes que não são comuns aos dois pais, são escolhidos aqueles dos pais com o maior *fitness* como exemplificado na Figura 3.

O NEAT utiliza a especiação para resolver o problema de como proteger inovações topológicas que precisam de tempo para serem otimizadas de forma com que elas não sejam excluídas da população prematuramente. Quando um novo nó ou conexão são criados o *fitness* do organismo tende a cair pois tais inovações ainda não tiveram tempo de serem otimizadas (e.g. terem os pesos otimizados). A especiação divide a população em nichos (e.g. espécies), agrupando organismos semelhantes para evitar que organismos com

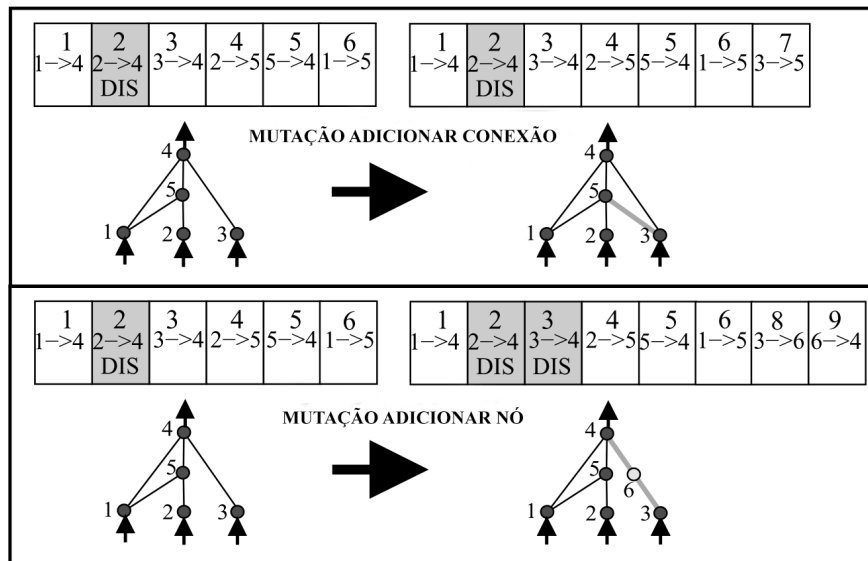


Figura 2. Exemplo da mutação para adição de novos nós ou conexões a topologia retirada do artigo de [Stanley and Miikkulainen 2002]

novas inovações sejam comparados com toda a população. Para agrupar os organismos semelhantes, o NEAT utiliza os marcadores de inovação. A semelhança entre organismos é calculada baseado em quantas inovações em comum eles compartilham, então um *threshold* é utilizado para dizer se o organismo é de tal espécie.

Para minimizar a topologia da RNA sem utilizar métricas de complexidade na função objetivo, o NEAT parte de organismos simples, com zero nós na camada escondida. Então os organismos vão ficando mais complexos incrementalmente com o passar das gerações, adicionando mais nós e conexões. Se a conexão ou o nó adicionado representar um comportamento útil, ele será mantido pelas gerações. Desta forma o NEAT garante uma busca que minimiza a complexidade da estrutura das soluções, e que tenta maximizar a taxa de acerto na classificação.

4. Metodologia Experimental

Neste trabalho, o desempenho do NEAT foi avaliado em três bases de dados diferentes: a *Image Retrieval in Medical Applications* (IRMA), contendo imagens de mamografia; a base do *International Skin Imaging Collaboration* (ISIC) [Gutman et al. 2016], contendo imagens dermatológicas; e a base de dados *Wisconsin Breast Cancer Diagnostic* (WDBC) [Wolberg et al. 1992], retirada do repositório de base de dados da *UCI Repositories*.

Inicialmente, o NEAT foi comparado com quatro classificadores, sendo eles o MLP, RBF, SVM e o *Random Forest*. Estes quatro algoritmos são largamente utilizados na literatura em diferentes problemas de classificação, sendo por isto escolhidos neste experimento. A função de objetivo do NEAT utilizada foi a taxa de acerto das predições corretas realizadas pela RNA. Os algoritmos foram contrastados através da comparação de suas taxas de precisão obtidas em bases de imagens médicas. Foram consideradas neste primeiro estudo, as duas primeiras bases de dados, a IRMA e a ISIC. Vale salientar que foi utilizado um subconjunto de instâncias dessas bases, considerando-se 2796 instâncias

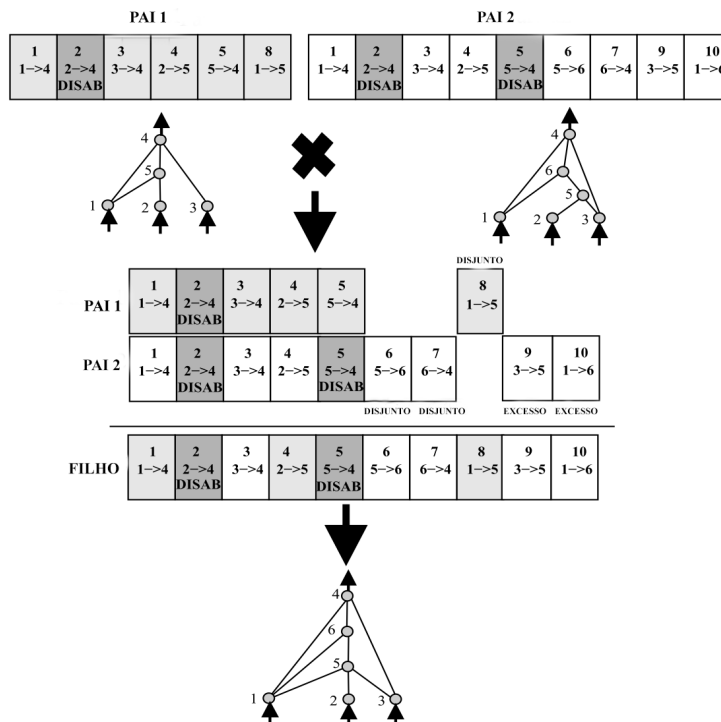


Figura 3. Exemplo de cruzamento entre dois organismos retirado do artigo de [Stanley and Miikkulainen 2002]

da base IRMA e 200 instâncias da base ISIC. A base de imagens IRMA foi testada utilizando três descritores diferentes, sendo estes o *Gray Level Co-Occurrence Matrices* (GLCM) [Haralick et al. 1973], extraindo 13 características, e o *Local Binary Patterns* (LBP), extraindo 26 características e o Zernike que extrai 25. As imagens são rotuladas em três classes: normal, benigno e maligno. Na base ISIC foram utilizados os descritores GLCM e o LBP, sendo as imagens classificadas entre lesões benignas e malignas. Foi realizada a normalização das bases de dados para execução dos experimentos.

O NEAT também foi avaliado na base de dados WDBC (com 569 instâncias), pelo fato dela ser utilizada por vários dos trabalhos relacionados (ver seção 2). O NEAT foi comparado com três das abordagens citadas: a MLP/GA Híbrida, proposta por [Belciug and Gorunescu 2013], a CGPANN-RBF *crossover* proposta por [Manning and Walsh 2013] e a CGPANN, proposta por [Ahmad et al. 2012].

Para garantir a confiabilidade dos resultados obtidos, todos os experimentos com o NEAT foram executados 30 vezes e a média dos resultados foi utilizada. A Tabela 1 mostra os parâmetros utilizados no NEAT para cada uma das bases utilizadas. Esses parâmetros foram definidos empiricamente, após vários experimentos realizados utilizando as bases de dados com os descritores citados. Os experimentos feitos com os outros algoritmos de classificação foram executados no programa WEKA [Hall et al. 2009] utilizando os parâmetros padrão de cada algoritmo e utilizando a validação cruzada com 10 folds. Embora os algoritmos não sejam executados com seus parâmetros ótimos, os resultados obtidos por estes algoritmos com seus parâmetros *default* já mostraram ser muito

bons em diferentes problemas de classificação [Mantovani et al. 2015]. O objetivo, portanto, é utilizar os classificadores como *baseline* e verificar se o NEAT têm a capacidade de superá-los.

Tabela 1. Parâmetros utilizados no NEAT

Base	Número de Gerações	Taxa de Adição de Nós	Taxa de Adição de Conexões	Limiar de compatibilidade
IRMA com GLCM	3.000	0,01	0,1	6
IRMA com LBP	3.000	0,01	0,1	6
IRMA com Zernick	3.000	0,01	0,1	6
ISIC com GLCM	10.000	0,01	0,1	7,5
ISIC com LBP	10.000	0,01	0,1	7,5
WDBC	1.000	0,01	0,1	7,5

5. Resultados

Nesta seção serão discutidos os resultados obtidos a partir da comparação do NEAT com a MLP, RBF, SVM e *Random Forest*. Três análises foram conduzidas, uma utilizando a base de imagens de mamografia IRMA, outra utilizando a base de imagens dermatológicas ISIC. Os experimentos com a IRMA foram realizados utilizando os descritores GLCM, LBP e Zernike. Uma última análise foi realizada utilizando a base de dados WDBC. Os experimentos com a base ISIC utilizaram os descritores GLCM e LBP.

5.1. Base de dados IRMA

A Tabela 2 apresenta a taxa de acerto obtida por cada algoritmo variando-se o descritor. Como o NEAT foi executado 30 vezes, por ser um algoritmo estocástico, também é apresentado o seu desvio padrão.

Como se pode ver na Tabela 2, o NEAT obteve bons resultados quando comparado com os outros algoritmos. Em relação ao descritor GLCM, o NEAT obteve o terceiro melhor resultado (61,166%) dos cinco algoritmos, sendo superado pela MLP e *Random Forest*. Quanto ao descritor LBP, o NEAT obteve o segundo melhor resultado (58,121%), perdendo apenas para o MLP. Por fim, o NEAT obteve o melhor resultado (53,349%) em relação aos demais algoritmos, no descritor Zernick. Em média o NEAT conseguiu uma precisão 7,15% melhor do que os outros algoritmos.

Tabela 2. Resultados dos experimentos na base IRMA

	NEAT	MLP	RBF	SVM	Random Forest
GLCM	61,166 ($\pm 0,01488$)	67,096	58,405	43,920	65,987
LBP	58,121 ($\pm 0,04746$)	59,871	42,919	37,125	55,186
Zernick	53,349 ($\pm 0,01490$)	43,527	39,628	45,887	45,1

5.2. Base de dados ISIC

A Tabela 3 apresenta a taxa de acerto obtida por cada algoritmo variando-se o descritor. Na base de imagens ISIC, o NEAT obteve resultados superiores a todos os algoritmos independente do descritor utilizado, chegando a alcançar uma precisão máxima de 83,5% e média de 78,5% com o descritor GLCM. Quanto ao descritor LBP, o NEAT obteve precisão média de 77,883%, superando com folga os demais classificadores. Além disso, também se pode ver que o NEAT obteve em média uma precisão 18,47% melhor para o descritor GLCM e 16,88% melhor para o descritor LBP.

Tabela 3. Resultados dos experimentos na base ISIC

	NEAT	MLP	RBF	SVM	Random Forest
GLCM	78,850 ($\pm 0,02707$)	61,5	65	49,5	65,5
LBP	77,883 ($\pm 0,02696$)	65	56,5	51,5	71

5.3. Base de dados WDBC

A Tabela 4 apresenta a comparação da taxa de acerto do NEAT com a de outros algoritmos de neuroevolução propostos na literatura. O NEAT apresentou resultados semelhantes aos dos algoritmos, ficando na frente da MLP/GA Híbrida de [Belciug and Gorunescu 2013] (com 100.000 gerações). Porém teve resultados em média 3,76% piores do que os outros algoritmos. No entanto, vale salientar que enquanto o CGPANN de [Ahmad et al. 2012] e o CGPANN-RBF de [Manning and Walsh 2013] executaram 100.000 gerações, o NEAT alcançou um resultado muito semelhante com apenas 1.000 gerações.

Tabela 4. Comparação dos resultados do NEAT com os dos trabalhos relacionados na base de dados WDBC

Algoritmo	Resultado	Referência
NEAT	93,714 ($\pm 0,00275$)	-
MLP/GA Híbrida	93,58	[Belciug and Gorunescu 2013]
CGPANN-RBF <i>crossover</i>	97,19	[Manning and Walsh 2013]
CGPANN	96	[Ahmad et al. 2012]

6. Conclusão e Trabalhos Futuros

Este trabalho faz um estudo experimental da eficiência do algoritmo de Neuroevolução com Aumento Topológico (NEAT) aplicado a detecção de câncer em imagens médicas. Neste estudo, o NEAT foi comparado com dois tipos de redes neurais artificiais MLP, RBF, e os algoritmos *Random Forest* e SVM. O NEAT apresentou resultados promissores, alcançando uma precisão superior aos algoritmos de classificação MLP, RBF, SVM e *Random Forest* em quase todos os descritores utilizados, em diferentes bases de imagens médicas. O NEAT também apresentou resultados semelhantes ao de algoritmos de neuroevolução propostos na literatura, só que com um número de iterações 100 vezes menor que dos demais algoritmos.

Como trabalhos futuros, pretende-se selecionar os hiper-parâmetros do NEAT de forma sistemática, evitando-se, assim, a escolha manual destes por tentativa e erro. O objetivo é fazer com que o NEAT seja ainda mais capaz de alcançar melhores resultados na classificação de imagens. Além disso pretendemos otimizar RNAs com o NEAT levando-se em consideração outras funções objetivo. Também pretende-se estudar o efeito da seleção de características. A seleção de características é feita para escolher quais características da base dados serão utilizadas, pois esta tarefa pode ter um grande efeito na eficiência do algoritmo, diminuindo o espaço de busca.

Referências

Ahmad, A. M., Khan, G. M., Mahmud, S. A., and Miller, J. F. (2012). Breast cancer detection using cartesian genetic programming evolved artificial neural networks. pages 1031–1038.

- Ahmad, F., Isa, N. A. M., Hussain, Z., Osman, M. K., and Sulaiman, S. N. (2015). A ga-based feature selection and parameter optimization of an ann in diagnosing breast cancer. *Pattern Analysis and Applications*, 18(4):861–870.
- Belciug, S. and Gorunescu, F. (2013). A hybrid neural network/genetic algorithm applied to breast cancer detection and recurrence. *Expert Systems*, 30(3):243–254.
- Bhardwaj, A. and Tiwari, A. (2015). Breast cancer diagnosis using genetically optimized neural network model. *Expert Systems with Applications*, 42(10):4611–4620.
- Dheebea, J., Singh, N. A., and Selvi, S. T. (2014). Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach. *Journal of biomedical informatics*, 49:45–52.
- Gutman, D., Codella, N. C., Celebi, E., Helba, B., Marchetti, M., Mishra, N., and Halpern, A. (2016). Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1605.01397*.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18.
- Haralick, R. M., Shanmugam, K., et al. (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621.
- Ibrahim, A. O., Shamsuddin, S. M., yahya Saleh, A., Abdelmaboud, A., and Ali, A. (2015). Intelligent multi-objective classifier for breast cancer diagnosis based on multilayer perceptron neural network and differential evolution. pages 422–427.
- Manning, T. and Walsh, P. (2013). Improving the performance of cgpnn for breast cancer diagnosis using crossover and radial basis functions. pages 165–176.
- Mantovani, R. G., Rossi, A. L., Vanschoren, J., Carvalho, A. C. P. d. L., et al. (2015). Meta-learning recommendation of default hyper-parameter values for svms in classifications tasks. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases; International Workshop on Meta-Learning and Algorithm Selection*. University of Porto.
- Oliveiraa, J. E., Guelda, M. O., Araújo, A. d. A., Ottc, B., and Deserno, T. M. (2008). Towards a standard reference database for computer-aided mammography. 6915:69151Y.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127.
- Turabieh, H. (2016). Comparison of NEAT and Backpropagation Neural Network on Breast Cancer Diagnosis. *International Journal of Computer Applications*, 139(8):40–44.
- Wolberg, W. H., Street, W. N., and Mangasarian, O. L. (1992). Breast cancer wisconsin (diagnostic) data set. *UCI Machine Learning Repository* [<http://archive.ics.uci.edu/ml/>].
- Zohra, B. F. and Nacéra, B. (2013). Detection of tumor in mammographic images by rbf neural network and multi population genetic algorithm. *International Journal of Applied Information Systems (IJ AIS)*, 6(3).

Seleção Adaptativa de Operador da Evolução Diferencial para o Problema da Predição Terciária da Estrutura de Proteínas

Isaque Silverio¹, Richard A. Gonçalves¹, Carolina P. Almeida¹,
Myriam R. Delgado², Sandra M. Venske¹

¹Universidade Estadual do Centro-Oeste - UNICENTRO
Departamento de Ciência da Computação
Guarapuava, PR, Brasil

²Universidade Tecnológica Federal do Paraná - UTFPR
CPGEI/DAINF
Curitiba, PR, Brasil

silverio.isaque@gmail.com, myriamdelg@utfpr.edu.br

{richard, carol, ssvenske}@unicentro.br

Abstract. *The Protein Structure Prediction Problem (PSPP) is an important research topic in Bioinformatics, whose solution is still open. Knowing a protein's structure can help discovering what kind of activity the protein carries out in the cell and creating drugs to fight illnesses caused by its malfunctioning. The Differential Evolution (DE) is an evolutionary algorithm which represents an interesting alternative to deal with PSPP. DE is highly dependent on its parameters. Probability Matching (PM) and Adaptive Pursuit (AP) are techniques that can be used to adaptively select DE operators. In this work, PM and AP were used to perform an adaptive selection of DE operators for the PSPP.*

Resumo. *O Problema da Predição da Estrutura de Proteínas (PPEP) é um tópico importante de pesquisa em Bioinformática, cuja solução ainda está em aberto. Conhecer a estrutura de uma proteína pode ajudar a descobrir que tipo de atividade a proteína realiza na célula e criar drogas para combater doenças causadas pelo seu mau funcionamento. A Evolução Diferencial (ED) é um algoritmo evolucionário que representa uma alternativa interessante para abordar o PPEP. A ED é altamente dependente dos seus parâmetros. O Probability Matching (PM) e o Adaptive Pursuit (AP) são técnicas que podem ser utilizadas para seleção adaptativa de parâmetros. Neste trabalho, propõe-se a utilização do PM e AP para a seleção adaptativa do operador de seleção de estratégias da ED aplicada ao PPEP.*

1. Introdução

O Problema da Predição da Estrutura de Proteínas (PPEP) é um problema de minimização da energia potencial livre de uma proteína e um tópico importante de pesquisa, ainda sem solução, em Bioinformática. Conhecendo a estrutura de uma proteína, os cientistas podem descobrir que tipo de atividade a proteína realiza na célula e criar drogas para combater doenças relacionadas ao seu mau funcionamento.

Na literatura há várias abordagens para modelagem computacional do PPEP [Tramontano 2006]. Neste trabalho a modelagem utilizada é a *ab initio*, que prevê a conformação nativa de uma proteína considerando somente sua sequência de aminoácidos. Esta predição se mostra um teste fundamental da compreensão do enovelamento da proteína e um desafio prático importante, tendo em vista que está sendo produzido um grande número de sequências de proteínas sem que suas informações 3D sejam conhecidas [Tramontano 2006]. A metodologia *ab initio* requer três elementos [Osguthorpe 2000]: uma representação da geometria da proteína, uma função potencial e uma técnica de busca no espaço da superfície da energia livre.

A *representação da geometria* se dá pelo cálculo da energia potencial, a fim de procurar o espaço para o seu enovelamento. Neste trabalho foi utilizada a representação livre de reticulado (*off-lattice*), em que uma proteína é representada como uma cadeia de resíduos de aminoácidos (ou grupos de resíduos) que se deslocam em um espaço contínuo, utilizando suas coordenadas atômicas. A *função potencial* retorna um valor para a energia de acordo com a sua conformação. Aqui foi investigado o uso da função CHARMM (v.27) para calcular a função de energia. Nas *técnicas de buscas conformacionais* são calculadas as energias envolvidas no processo do enovelamento para encontrar a estrutura com a energia livre mais baixa. Neste trabalho foi utilizada a Evolução Diferencial (ED) como técnica de busca.

A ED é um algoritmo eficiente que, geralmente, apresenta rápida convergência na busca das soluções desejadas em problemas de otimização contínua. A Evolução Diferencial tem como base alguns parâmetros como tamanho da população (NP), constante de cruzamento (CR) e constante de mutação (F), além de estratégias de mutação.

A adaptação de parâmetros tem como objetivo superar as desvantagens da ED clássica, em que a definição de valores adequados pode levar a elevados custos computacionais, devido a processos de “tentativa-e-erro” demorados e que ainda podem ser ineficazes nos casos em que os valores dos parâmetros ideais variam durante o processo evolutivo. No presente trabalho, os parâmetros NP , CR e F são fixos e a escolha da estratégia de mutação foi realizada por meio da utilização de duas técnicas adaptativas, testadas separadamente: *Probability Matching* e *Adaptive Pursuit* [Goldberg 1990].

Assim, propõe-se aqui o desenvolvimento e teste de um algoritmo a ser aplicado para o problema da predição da estrutura de proteínas, utilizando o método de evolução diferencial. A escolha da estratégia de mutação é aplicada por meio das técnicas de adaptação *Probability Matching* e *Adaptive Pursuit*. A abordagem considerada é mono-objetivo.

Este trabalho está estruturado como segue. A Seção 2 fornece alguns conceitos básicos sobre o problema abordado, a evolução diferencial e técnicas adaptativas utilizadas. Na Seção 3 é apresentado o algoritmo proposto. Na Seção 4 são mostrados os resultados obtidos e sua análise. Finalmente, as conclusões e os trabalhos futuros são discutidos na Seção 5.

2. Revisão Bibliográfica

Esta seção contém um breve levantamento sobre os principais conceitos utilizados, incluindo o problema da predição da estrutura de proteínas, a técnica da evolução diferencial e técnicas adaptativas utilizadas. Alguns trabalhos relacionados também são discutidos.

2.1. Problema da Predição da Estruturas das Proteínas (PPEP)

A Predição da Estrutura das Proteínas, pode ser tido como um dos problemas mais difíceis na Bioinformática na atualidade. Este é um problema complexo que faz parte do grupo de problemas NP-difíceis [Hart and Istrail 1997], sendo que uma solução eficiente para o mesmo ainda está em aberto.

O PPEP é um problema de minimização, com o objetivo de encontrar a estrutura terciária com a energia mínima de uma proteína. Das milhares de proteínas existentes, apenas se conhece uma pequena parcela das estruturas que as compõem. O conhecimento das estruturas das proteínas é de grande importância, pois com essas informações é possível descobrir a função que a proteína irá exercer na célula. Isto pode auxiliar os cientistas na descoberta de novas medicações ou até na cura de doenças como Alzheimer, alguns tipos de câncer, Parkinson, entre outras; já que algumas doenças estão ligadas ao mau funcionamento das proteínas.

Existem várias abordagens que auxiliam no processo de predição da estrutura tridimensional de proteínas, estas abordagens podem ser divididas em dois grupos: baseadas em padrões (homologia e *threading*) e os sem padrões (*ab initio* e *de novo*) [Tramontano 2006].

Ao contrário da modelagem por homologia, os métodos *ab initio* não utilizam informações de bases de dados estruturais de proteínas. Eles são fundamentados na observação que a estrutura nativa de uma proteína corresponde a uma estrutura na qual a energia livre é mínima. O método tem como objetivo encontrar o valor mínimo global da energia livre de uma proteína que corresponderia, ou à estrutura nativa ou a uma conformação funcional da mesma [Osguthorpe 2000]. A principal vantagem desse método é que ele é capaz de prever novos enovelamentos de proteínas, visto que não está limitado ao uso de estruturas já conhecidas.

Neste trabalho utilizou-se a representação livre de reticulado (*off-lattice*) para a representação da proteína. Nela, uma proteína é representada como uma cadeia de resíduos de aminoácidos (ou grupos de resíduos) que se movem em um espaço contínuo, utilizando suas coordenadas atômicas. Cada tipo de resíduo tem um número pré-estabelecido de ângulos de torção para alcançar a conformação da proteína. A espinha dorsal de cada resíduo é representada por 3 ângulos: ϕ , ψ , ω . As cadeias laterais são representadas pelos ângulos χ . O número total de ângulos χ depende do tipo de resíduo (conforme Tabela 1). Os valores dos ângulos podem variar entre -180° a 180° .

Tabela 1. Número de ângulos χ em cada resíduo de aminoácido.

Resíduo de aminoácido	Número de ângulos χ
GLY, ALA, PRO	cadeia principal
SER, CYS, THR, VAL	χ_1
ILE, LEU, ASP, ASN, HIS, PHE, TYR, TRP	χ_1, χ_2
MET, GLU, GLN	χ_1, χ_2, χ_3
LYS, ARG	$\chi_1, \chi_2, \chi_3, \chi_4$

A representação *ab initio* oferece uma perspectiva mais próxima da realidade do processo de dobramento da estrutura da proteínas, pois estas podem se movimentar livremente no espaço conformacional.

Existem muitos campos de força (funções de energia potencial) para a simulação

de proteínas com bons resultados na literatura. Aqui foi utilizado o CHARMM, cuja formulação obedece a Equação 1. Ela fornece o valor aproximado da energia potencial de uma conformação de proteína.

Os componentes de ligação, ângulos e diedrais representam a energia dos átomos ligados, enquanto a última parte da fórmula representa a energia dos átomos não ligados. Em uma abordagem multiobjetivo, a otimização destas energias é realizada separadamente, diferente da formulação mono-objetivo (utilizada neste trabalho) que considera a soma das energias.

$$E_C = \sum_{\text{ligacoes}} K_b(b_C - b_{eq})^2 + \sum_{\text{angulos}} K_\theta(\theta_C - \theta_{eq})^2 + \sum_{\text{diedrais}} K_\phi[1 + \cos(n\Phi_C - \gamma)] + \sum_{\text{atomos nao-ligados}} \left[\frac{A_{ij}}{r_{ij,C}^{12}} - \frac{B_{ij}}{r_{ij,C}^6} + \frac{q_i q_j}{\varepsilon_0 r_{ij,c}} \right] \quad (1)$$

tal que: K_b é a constante de ligação; K_θ é a constante para ângulos; K_ϕ é a constante para ângulos de torção (diedrais); b é a distância de ligação entre dois átomos; b_{eq} é a distância ideal entre dois átomos; θ é o ângulo da ligação química; θ_{eq} é o ângulo ideal da ligação química; n é o número de mínimos; γ é o deslocamento angular; ε é a energia de atração/repulsão de Lennard-Jonnes; A e B são constantes para o potencial de Lennard-Jonnes; q_i e q_j são as cargas de átomos não-ligados; r é a distância entre átomos não-ligados; ε_0 é a constante dielétrica.

Nas técnicas de buscas conformacionais são calculadas as energias envolvidas no processo do enovelamento para encontrar a estrutura com a energia livre mais baixa. O método evolucionário da Evolução Diferencial será brevemente descrito na Seção 2.2 .

2.2. Evolução Diferencial (ED)

A ED é um método de otimização com um excelente desempenho numa grande variedade de problemas de otimização [Plagianakos et al. 2008]. Proposta inicialmente na década de 1990, a ED foi projetada para a otimização de sistemas com variáveis contínuas. Um dos motivos principais do sucesso da ED está relacionado principalmente ao seu poderoso mas simples mecanismo de busca, o qual usa a diferença entre dois vetores, escolhidos aleatoriamente entre as soluções candidatas, para produzir novas soluções. À medida que a população evolui, a direção e o tamanho do passo da busca na mutação mudam, ajustando-se de acordo com a distribuição da população no espaço de busca.

A principal ideia da evolução diferencial é primeiramente criar uma população inicial aleatória, composta por NP indivíduos espalhados sobre o espaço de busca e avaliar esses indivíduos. Em seguida, a população é modificada a cada iteração por meio de três operadores da ED sendo eles: mutação (F), cruzamento (CR) e seleção para chegar na próxima geração.

Primeiramente, na **mutação** são selecionados indivíduos aleatórios da população (*vetores diferença*) e na sequência é calculada a diferença ponderada entre esses in-

divíduos. A esta diferença é aplicado o **fator de mutação** gerando um indivíduo doador (*vetor doador*) e por fim é feita a adição do indivíduo doador com um indivíduo da população (*vetor alvo*), assim gerando o indivíduo experimental (*vetor experimental*).

Na **operação de cruzamento** os componentes do indivíduo experimental são misturados com os componentes de um indivíduo pai (vetor), assim gerando um indivíduo descendente (vetor descendente). A ED possui dois principais tipos de cruzamento sendo eles *cruzamento binomial* e *cruzamento exponencial*. Neste trabalho foi adotado o *cruzamento binomial* descrito pela Equação 2 .

$$x'_{i,j} = \begin{cases} u_{i,j}, & \text{se } r_j \leq CR \text{ ou } j = d \\ x_{i,j}, & \text{caso contrário} \end{cases} \quad (2)$$

A Equação 2 gera o vetor descendente x'_i por meio da cópia dos componentes do vetor experimental u_i e do vetor pai x_i , dependendo de um número aleatório $r_j \in [0, 1]$ que é comparado ao parâmetro CR . Devido ao fato que durante o processo de seleção o vetor pai e o vetor descendente irão competir para compor a geração subsequente, é assegurado que $x'_i \neq x_i$ pela seleção de pelo menos um componente do vetor experimental u_i . Para este propósito a variável $d \in \{1, \dots, n\}$ (n é a dimensão do problema) é escolhida aleatoriamente para cada vetor descendente em cada geração e, desta forma, garante-se que o j -ésimo componente do vetor descendente seja copiado do vetor experimental (somente um componente) [Storn and Price 1997].

No processo de **seleção** da ED é realizado um teste sobre o indivíduo descendente, em que se verifica se o *fitness* do indivíduo descendente é melhor que o indivíduo pai, caso tal condição seja verdadeira o indivíduo descendente será guardado na população para a próxima geração, caso contrário, o indivíduo pai se mantém na próxima geração.

As fases de mutação, cruzamento e seleção compõem as estratégias da evolução diferencial. Existem várias estratégias diferentes que podem ser aplicadas à ED, sendo elas apresentadas pela notação “ $ED/x/y/z$ ”. Tais estratégias diferem na forma em que o vetor alvo é encontrado (x), no número de vetores de diferença a ser usado (y) e, por fim, no tipo de cruzamento utilizado (z) [Storn and Price 1997]. Neste trabalho foram utilizadas as estratégias apresentadas na Tabela 2.

Tabela 2. Estratégias de mutação da ED utilizadas.

Estratégia	Notação
$ED/rand/1$	$\omega = x_1 + F * (x_2 - x_3)$
$ED/rand/2$	$\omega = x_1 + F * (x_2 + x_3 - x_4 - x_5)$
$ED/non - linear$	$\omega = w^2 \mathbf{c}_a + w \mathbf{c}_b + \mathbf{c}_c$

As estratégias $ED/rand/1$ e $ED/rand/2$ selecionam aleatoriamente o vetor alvo (x_1) e utilizam um ou dois pares de vetores diferença, respectivamente. A estratégia chamada aqui de $ED/Non - Linear$, foi proposta em [Sindhya et al. 2011] e é definida pela Equação 3.

$$\omega = w^2 \mathbf{c}_a + w \mathbf{c}_b + \mathbf{c}_c, \quad (3)$$

sendo que w é dado pela Equação 4.

$$w \in \begin{cases} U[0, 2], & \text{se } rand \leq P_{inter} \\ U[2, 3], & \text{caso contrário.} \end{cases} \quad (4)$$

com $rand \in U[0, 1]$ (i.e., $rand$ é um valor entre 0 e 1 aleatoriamente gerado por uma distribuição uniforme) e a probabilidade de interpolação, P_{inter} definida como sugerido em [Sindhya et al. 2011]. Os indivíduos c_a , c_b e c_c são definidos de acordo com a Equação 5.

$$\begin{aligned} c_a &= \frac{\mathbf{x}_c - 2\mathbf{x}_b + \mathbf{x}_a}{2}, \\ c_b &= \frac{4\mathbf{x}_b - 3\mathbf{x}_c - \mathbf{x}_a}{2}, \\ c_c &= \mathbf{x}_c \end{aligned} \quad (5)$$

tal que \mathbf{x}_a , \mathbf{x}_b e \mathbf{x}_c são indivíduos aleatoriamente selecionados da população atual.

2.3. Técnica Adaptativa: *Probability Matching (PM)*

Probability Matching [Goldberg 1990] é uma técnica em que se visa calcular a probabilidade de uso de cada estratégia de um determinado conjunto de estratégias proposto. Esse cálculo é realizado com base em quatro parâmetros: médias das melhorias dos valores de *fitness* dos indivíduos, média de qualidade das soluções, taxa de adaptação e um valor de probabilidade de uso das estratégias.

Para a avaliação de desempenho de um determinado conjunto de estratégias, PM utiliza uma medida de *recompensa*, em que se utiliza o grau de melhoria no valor de *fitness*. Se houver alguma melhoria nesse valor, então é atribuído, à estratégia associada àquele indivíduo, um valor numérico como recompensa.

Seja um conjunto de estratégias $\zeta = \{s_1, \dots, s_S\}$ e $S > 1$ e um vetor de probabilidades $\mathbf{p}(Ger) = (p_1(Ger), \dots, p_S(Ger))$, tal que $p_s(Ger)$ é a probabilidade de que a s^{esima} estratégia seja utilizada na geração Ger . Existem vários métodos para utilizar adaptativamente a probabilidade $p_s(Ger)$ da s^{esima} estratégia [Thierens 2005]. Em todos os casos tem-se que $\sum_{s=1}^S p_s(Ger) = 1$; $\forall Ger : p_{min} \leq p_s(Ger) \leq 1$, sendo p_{min} o valor de probabilidade mínimo que cada estratégia possui, tal valor é utilizado para que nenhuma estratégia deixe de ser considerada ao longo do processo de busca [Thierens 2005].

O método *Probability Matching* utiliza $p_s(Ger)$ conforme a Equação 6 [Thierens 2005]:

$$p_s(Ger+1) = p_{min} + (1 - S * p_{min}) * \frac{q_s(Ger+1)}{\sum_{s=1}^S q_s(Ger+1)} \quad (6)$$

na qual

$$q_s(Ger+1) = q_s(Ger) + \alpha * [r_s(Ger) - q_s(Ger)] \quad (7)$$

tal que a taxa de adaptação é $\alpha \in (0, 1]$, $q_s(Ger)$ é a qualidade empírica e $r_s(Ger)$ representa a recompensa que a s^{esima} estratégia recebe.

Na Equação 6 percebe-se que $\sum_{s=1}^S p_s(Ger+1) = 1$. Quando apenas uma estratégia obtém recompensa por um longo tempo, a sua probabilidade de seleção converge $p_{max} = p_{min} + (1 - S * p_{min})$.

O mecanismo de recompensa funciona com o melhoramento do indivíduo da população durante as avaliações, em relação ao pai e ao melhor indivíduo, como mos-

tra a Equação 8.

$$\eta_{s(Ger)} = \begin{cases} f_{best}/f_{c(Ger)} * |f_{p(Ger)} - f_{c(Ger)}|, & \text{Se } f_{c(Ger)} \text{ melhor que } f_{p(Ger)} \\ 0 & \text{caso contrário,} \end{cases} \quad (8)$$

sendo que f_{best} é o valor de *fitness* da melhor solução da população, $f_{p(Ger)}$ e $f_{c(Ger)}$ são, respectivamente, os valores de *fitness* do vetor pai e seu filho na geração Ger , ambos associados com um melhoramento. Caso não ocorra nenhum melhoramento, ou seja, o filho é igual ou pior que o pai, o impacto da estratégia é considerado nulo ($\eta_{s(Ger)}^u = 0$).

No final de cada geração Ger , um valor de recompensa $r_{s(Ger)}$ é usado para atualizar a medida de qualidade $q_{s(Ger+1)}$ das estratégias. Esse valor é obtido pela Equação 9 [Thierens 2005]. A equação é a recompensa conhecida como *Absolute Reward* [Thierens 2005].

$$r_{s(Ger)} = \eta_{s(Ger)}^* = \max[\eta_{s(Ger)}^u]_{u=1, \dots, |R_s|} \quad (9)$$

com $|R_s|$ sendo a cardinalidade R_s (sendo $R_{s(Ger)} = \{\eta_s^1(g) \dots \eta_s^{|R_s|}(g)\}$ o conjunto de todos os melhoramentos encontrados adotando a s^{esima} estratégia na geração Ger).

2.4. Técnica Adaptativa: *Adaptive Pursuit* (AP)

Os algoritmos *Pursuit* são uma classe de algoritmos de rápida convergência propostos inicialmente por Thathachar e Sastry [Thathachar and Sastry 1984] para o aprendizado de autômatos. A estratégia *Adaptive Pursuit* (AP) aplicada ao algoritmo de seleção de operadores aqui proposto busca adaptar o vetor de probabilidades $p_{s(Ger)}$ de forma a maximizar o uso da s^{esima} estratégia que possui a maior qualidade $q_{s(Ger)}$.

No método AP, assim como no PM, o algoritmo de alocação de operadores seleciona uma estratégia s para ser executada de acordo com a probabilidade $p_{s(Ger)}$, e atualiza a correspondente qualidade $q_{s(Ger)}$ da estratégia com a recompensa recebida (Equação 7). O que difere as duas estratégias é como as probabilidades são adaptadas. Na estratégia AP, após a atualização do vetor qualidade, a estratégia s^* com maior qualidade $q_{s(Ger)}$ é considerada como sendo a melhor estratégia para o estágio atual do processo e sua probabilidade $p_{s(Ger)}$ é incrementada de acordo com a Equação 10. Também é preciso adicionar um valor mínimo P_{min} . Sendo assim, os valores das probabilidades ficarão dentro do intervalo $[P_{min}, P_{max}]$, sendo $0 < P_{min} < P_{max} < 1$.

$$\begin{aligned} p_{s(Ger+1)} &= p_{s(Ger)} + \beta[p_{max} - p_{s(Ger)}] \\ \forall s \neq s^* : p_{s(Ger+1)} &= p_{s(Ger)} + \beta[P_{min} - p_{s(Ger)}] \end{aligned} \quad (10)$$

sendo tal que se tem a seguinte restrição $P_{max} = 1 - (S - 1)P_{min}$. Para a relação $P_{max} > P_{min}$ ser verdadeira (desejável para que haja adaptação) é necessário respeitar a seguinte restrição: $P_{min} < 1/S$, sendo S o número total de estratégias consideradas.

2.5. Trabalhos relacionados

Vários trabalhos, como o aqui proposto, implementam a abordagem *ab initio* em ambientes *off-lattice* para o problema da predição da estrutura de proteínas.

Com abordagem mono-objetivo (como no presente trabalho), pode-se citar os seguintes trabalhos. Em [Liu and Tao 2006] uma proposta utilizando *Simulated Annealing* chamada PSAGC (*Parallel Simulated Annealing with Genetic Crossover*) é utilizada para minimizar a energia de proteínas para o PPEP. PSAGC usa ECEPP/2 para a

função de energia. Em [Nicosia and Stracquadanio 2007] é proposto um método chamado NOMAD-PSP (*Nonlinear Optimization for Mixed vAriables and Derivatives algorithm for PSP*). NOMAD-PSP baseia-se em dois algoritmos de busca direta: *Generalized Pattern Search* (GPS) e *Mesh Adaptive Direct Search* (MADS), a fim de encontrar a estrutura tridimensional com o menor valor da função energia possível. NOMAD-PSP utiliza o CHARMM como função de energia. Em [Sudha et al. 2013] o algoritmo adaptativo SaDE (*Self-adaptive Differential Evolution*) é utilizado em conjunto com o RGA (*Real-coded Genetic Algorithm*), que é um algoritmo inspirado pela teoria biológica da evolução. CR , F e estratégias de mutação são adaptados durante o processo evolucionário. As funções de energia utilizadas são ECEPP/2 e ECEPP/3.

Com abordagem multiobjetivo, pode-se citar os seguintes trabalhos. Em [Cutello et al. 2006] é apresentado um algoritmo imune sequencial para abordar o problema da predição da estrutura de proteínas, chamado IPAES (*Immune inspired Pareto Archived Evolutionary Strategy*). [Cutello et al. 2008] é uma extensão de [Cutello et al. 2006] onde os autores exploram as vantagens de sua abordagem a fim de validá-la utilizando um estudo mais detalhado com testes em um número maior de proteínas com a função de energia CHARMM. Em [Venske et al. 2016] os autores modelam o PPEP e aplicam ao algoritmo ADEMO/D (*Adaptive Differential Evolution for Multi-objective Problems based on Decomposition*). ADEMO/D incorpora conceitos de decomposição de problemas e mecanismos de adaptação de estratégias de mutação da ED em uma abordagem multiobjetivo.

3. Algoritmo Proposto

Nesta seção é apresentado o algoritmo proposto, detalhando seu funcionamento.

Nos primeiros passos do Algoritmo 1, nas linhas 1 a 4 são inicializados os parâmetros da ED, do método adaptativo e também ocorre a inicialização e avaliação da população inicial (conforme a Equação 1). Em seguida, o algoritmo entra no laço principal (linha 5 a 29), que seleciona a estratégia a ser usada na geração Ger . O conjunto das estratégias candidatas está incluído na Tabela 2.

No passo 7 é selecionada a estratégia a ser usada para gerar um novo indivíduo u_{ij} . A seleção é realizada na geração atual Ger , baseada na medida de probabilidade $p_s(Ger)$. Inicialmente, todas as estratégias candidatas do conjunto têm a mesma probabilidade de serem selecionadas. Após selecionar a estratégia (passo 7), no passo 8 são selecionados os indivíduos que irão fazer parte do cruzamento e mutação. Nos passos 10 a 16 será gerada uma nova solução u_{ij} .

O indivíduo filho gerado u_i é avaliado no passo 17. Se u_i é melhor que x_i (indivíduo pai), u_i irá substituir x_i na população (passo 18). A avaliação da melhoria do indivíduo u_i (passo 19) é obtida pela Equação 8. Se nenhuma melhoria for alcançada, $\eta_{s(Ger)}^u$ é considerado nulo (passo 22).

Ao fim da geração Ger , um valor de recompensa é calculado (passo 25) pela Equação 9. Esse valor de recompensa é realizado para atualizar a medida de qualidade da estratégia. Em seguida a medida de qualidade é atualizada pela Equação 7 (passo 26). No passo 27, os valores de probabilidade das diferentes estratégias são atualizados conforme o método testado (Equação 6 para o método PM ou Equação 10 para o método AP) a fim de serem utilizados na próxima geração.

Algorithm 1 Pseudocódigo do algoritmo adaptativo proposto.

```

1: Inicializar os parâmetros da ED, PM e AP
2: Gerar e avaliar a população inicial  $P^0 = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ,  $\mathbf{x}_i = (x_{i_1}, x_{i_2}, \dots, x_{i_n})$ 
3:  $Ger = 1$ .
4: Para cada estratégia  $a$ , atribuir  $q_s(Ger) = 0$  e  $p_s(Ger) = 1/S$ 
5: repita
6:   para  $i = 1$  para NP faça
7:     Selecionar a estratégia  $s$ , baseada na probabilidade
8:     Selecionar uniformemente aleatoriamente  $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$ 
9:      $j_{rand} = rndint(1, D)$ 
10:    para  $j = 1$  para o  $D$  faça
11:      se  $rndreal_j[0, 1) < CR$  ou  $j == j_{rand}$  então
12:         $u_{ij}$  é gerado de acordo com a estratégia  $s$ 
13:      senão
14:         $u_{ij} = x_{ij}$ 
15:      fim se
16:    fim para
17:    Avaliar o filho  $u_i$ 
18:    se  $f(u_i)$  é melhor ou igual o  $f(x_i)$  então
19:      Calcular melhoramento  $n_i$  (Eq. 8)
20:      Substituir  $x_i$  por  $u_i$ 
21:    senão
22:      Atribuir  $n_i = 0$ 
23:    fim se
24:  fim para
25:  Calcular a recompensa  $r_s(Ger)$  para cada estratégia (Eq. 9)
26:  Atualizar a qualidade  $Q_s(Ger)$  para cada estratégia (Eq. 7)
27:  Atualizar a probabilidade  $P_s(Ger)$  para cada estratégia com PM (Eq.6) ou AP (Eq. 10)
28:   $Ger = Ger + 1$ 
29: até os critérios de parada serem satisfeitos

```

4. Análise dos Resultados Obtidos

Esta seção apresenta a análise dos resultados obtidos com os testes do algoritmo adaptativo proposto, considerando os métodos PM e AP. Os testes consideraram 30 execuções independentes do algoritmo para cada método adaptativo. A instância real utilizada foi a proteína *Met-Enkephalin* (1PLW), composta por 5 aminoácidos (TYR-GLY-GLY-PHE-MET) e 75 átomos (dimensão do problema). A *Met-Enkephalin* tem sido usada como caso base para teste em diversos estudos para o PPEP [Cutello et al. 2006, Liu and Tao 2006, Anile et al. 2007, Nicosia and Stracquadanio 2007, Cutello et al. 2008, Sudha et al. 2013, Venske et al. 2016], por isso ela foi escolhida para ser utilizada aqui. A Tabela 3 apresenta os valores dos parâmetros utilizados.

Os critérios de avaliação utilizados para a comparação entre a proteína real e as conformações preditas obtidas pelo algoritmos foram energia potencial livre (calculada Equação 1) e a métrica RMSD. O RMSD (*Root Mean Square Deviation*) é uma medida que avalia a semelhança entre a estrutura predita e a estrutura real da proteína [Tramontano 2006]. A Equação 11 mostra como o RMSD é calculado. Os valores de

Tabela 3. Parâmetros do algoritmo proposto para o PPEP.

	Valores	Descrição
Parâmetros da ED		
NP	100	Tamanho da População.
CR	1,0	Cruzamento.
F	0,5	Fator de Mutação.
MAX-AV	500.000	Número Máximo de Avaliações.
Parâmetros do PM e AP		
α	0,3	Taxa de Adaptação.
β	0,8	Taxa de aprendizagem.
p_{min}	0,05	Probabilidade Mínima de cada Estratégia.
K	3	Número de Estratégias do Conjunto.

RMSD foram calculados considerando somente os carbonos alfa.

$$RMSD_{C_{\alpha}}(a, b) = \sqrt{\frac{\sum_{i=1}^{N_a} |r_{ai} - r_{bi}|^2}{N_a}} \quad (11)$$

sendo que r_{ai} e r_{bi} são as posições dos átomos i nas estruturas a e b e N_a é o número total de átomos da estrutura [Tramontano 2006].

Na Tabela 4 são apresentados os resultados obtidos com o método AP, o método PM, bem como três versões não adaptativas, considerando cada uma das estratégias isoladamente (*ED/rand/1*, *ED/rand/2*, *ED/non-linear*). A tabela contém os valores de energia obtidos referentes ao pior indivíduo (que possui o maior valor de energia), associado ao seu RMSD, e o melhor indivíduo (menor valor de energia), com seu respectivo valor de RMSD das 30 execuções realizadas. Também são apresentadas as médias das energias e dos RMSDs obtidos, com os respectivos valores de desvios padrão. Os valores de energia são apresentados em $kcal\ mol^{-1}$ e os valores de RMSD em Ângström (Å).

Tabela 4. Resultados obtidos pelo algoritmo proposto com o método PM, o método AP e versões não adaptativas.

Estratégia	Menor Energia / RMSD _{C_α}	Média Energia - Desvio Padrão	Média RMSD _{C_α} - Desvio Padrão
Adaptativa (PM)	-32,260 / 2,082	-24,375 ± 2,640	1,755 ± 0,116
Adaptativa (AP)	-22,070 / 1,720	-15,590 ± 2,660	1,714 ± 0,133
<i>ED/Rand/1</i>	-10,953 / 1,785	-3,973 ± 2,890	1,669 ± 0,137
<i>ED/Rand/2</i>	-5,939 / 1,550	7,229 ± 5,769	1,689 ± 0,176
<i>ED/Non-linear</i>	-27,881 / 1,763	-26,678 ± 0,573	1,721 ± 0,058

Pode ser observado na Tabela 4 que a versão adaptativa utilizando o método PM obteve melhores resultados em termos de minimização de energia que o método AP. Além disso, o método *Probability Matching* também apresentou o melhor valor de média das energias em comparação com o AP. Apesar da energia potencial ser melhor no método *Probability Matching*, o *Adaptive Pursuit* apresentou uma conformação mais próxima da proteína real, pois o valor de RMSD associado foi menor. Além disso, o AP também apresentou uma pequena vantagem na média dos valores de RMSD das conformações preditas. Os valores de desvios padrão de ambos os métodos se mantiveram próximos.

Somente comparando as versões não-adaptativas, a versão que obteve a menor energia foi a que utiliza a estratégia *ED/Non-linear*, com uma pequena variação nas médias das energias totais encontradas e um bom valor de RMSD associado.

A versão adaptativa utilizando o método PM obteve o melhor valor de energia, quando comparada com a versão com o método AP e as outras versões não-adaptativas. Assim, essa foi a versão escolhida para o comparativo com a literatura (Tabela 5).

A Tabela 5 mostra os valores de energia e os valores de RMSD da proteína predita pelo algoritmo proposto utilizando PM e a proteína *Met-Enkephalin*. Os algoritmos comparados foram PSAGC [Liu and Tao 2006], NOMAD-PSP [Nicosia and Stracquadanio 2007], SaDE/RGA [Sudha et al. 2013], ADEMO/D [Venske et al. 2016] e IPAES [Cutello et al. 2006, Cutello et al. 2008].

Tabela 5. Resultados obtidos pelo algoritmo proposto e outras abordagens para *Met-Enkephalin*. Células em destaque representam os melhores valores.

Algoritmo	Energia (kcal mol ⁻¹)	RMSD _{C_α} (Å)
Algoritmo Proposto (PM)	-32,26	2,082
PSAGC	-11,10	-
NOMAD-PSP	-30,14	1,550
SaDE/RGA	-12,43	-
ADEMO/D	-24,68	1,681
IPAES	-20,56	1,740

Para a proteína *Met-enkephalin* o algoritmo proposto com o método PM obteve resultados competitivos considerando os valores de energia potencial. O valor de energia esteve associado a um bom valor de RMSD em comparação com a literatura.

5. Conclusões

Neste trabalho foi proposto um algoritmo usando o método da Evolução Diferencial adaptativo com as técnicas de adaptação *Probability Matching* e *Adaptive Pursuit* aplicado para o problema da predição da estrutura de proteínas.

A principal contribuição obtida foi o desenvolvimento de um otimizador com o propósito de avaliar o uso do método de evolução diferencial com dois métodos adaptativos para o problema da predição da estrutura de proteínas.

Os resultados alcançados com o algoritmo proposto utilizando o método *Probability Matching* foram melhores que os resultados com o método *Adaptive Pursuit*. Em geral, a abordagem adaptativa com PM teve um desempenho melhor quando comparada com versões em que cada uma das estratégias foi utilizada isoladamente. Os resultados encontrados também foram bons em comparação com a literatura, principalmente em termos de minimização nos valores de energia potencial.

Como trabalhos futuros pretende-se estudar outros métodos adaptativos, além de buscar a adaptação automática de outros parâmetros da ED.

Agradecimentos

Os autores agradecem à Fundação Araucária e ao CNPq (Projetos 456.179/2014-3, 483974/2013-7, 309197/2014-7) pelo apoio financeiro.

Referências

Anile, A. M., Cutello, V., Narzisi, G., Nicosia, G., and Spinella, S. (2007). Determination of protein structure and dynamics combining immune algorithms and pattern search methods. *Natural Computing*, 6(1):55–72.

- Cutello, V., Narzisi, G., and Nicosia, G. (2006). A multi-objective evolutionary approach to the protein structure prediction problem. *J R Soc Interface*, 3(6):139–51.
- Cutello, V., Narzisi, G., and Nicosia, G. (2008). Computational studies of peptide and protein structure prediction problems via multiobjective evolutionary algorithms. In Knowles, J., Corne, D., Deb, K., and Chair, D., editors, *Multiobjective Problem Solving from Nature*, Natural Computing Series, pages 93–114. Springer Berlin Heidelberg.
- Goldberg, D. E. (1990). Probability matching, the magnitude of reinforcement, and classifier system bidding. *Mach. Learn.*, 5:407–425.
- Hart, W. E. and Istrail, S. (1997). Robust proofs of np-hardness for protein folding: General lattices and energy potentials. *Journal of Computational Biology*, 4(1):1–22.
- Liu, Y.-L. and Tao, L. (2006). An improved parallel simulated annealing algorithm used for protein structure prediction. In *Machine Learning and Cybernetics, 2006 International Conference on*, pages 2335–2338.
- Nicosia, G. and Stracquadanio, G. (2007). Generalized pattern search and mesh adaptive direct search algorithms for protein structure prediction. In *Proceedings of the 7th International Conference on Algorithms in Bioinformatics, WABI'07*, pages 183–193, Berlin, Heidelberg. Springer-Verlag.
- Osguthorpe, D. J. (2000). Ab initio protein folding. *Current Opinion in Structural Biology*, 10(2):146–152.
- Plagianakos, V., Tasoulis, D., and Vrahatis, M. (2008). A review of major application areas of differential evolution. In *Advances in differential evolution*, pages 197–238. Springer.
- Sindhya, K., Ruuska, S., Haanpaa, T., and Miettinen, K. (2011). A new hybrid mutation operator for multiobjective optimization with differential evolution. *Soft Computing*, 15(10):2041–2055.
- Storn, R. and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. 11(4):341–359.
- Sudha, S., Baskar, S., and Krishnaswamy, S. (2013). Protein tertiary structure prediction using evolutionary algorithms. *International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)*, 3(3):338–348–595.
- Thathachar, M. and Sastry, P. (1984). A class of rapidly converging algorithms for learning automata.
- Thierens, D. (2005). An adaptive pursuit strategy for allocating operator probabilities. In *Conference on Genetic and Evolutionary Computation*, pages 1539–1546.
- Tramontano, A. (2006). *Protein Structure Prediction: Concepts and Applications*. John Wiley and Sons.
- Venske, S. M., Gonçalves, R. A., Benelli, E. M., and Delgado, M. R. (2016). Ademo/d: An adaptive differential evolution for protein structure prediction problem. *Expert Systems with Applications*, 56:209 – 226.

A Multimodal Recommendation Framework based on Low-dimensional Latent Factors

Anna Claudia Almeida¹, Anisio Lacerda¹,
Gisele Lobo Pappa², Flávio Luis Cardeal Pádua¹

¹Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas, 7675 – Nova Gameleira – Belo Horizonte – MG – Brazil

²Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 – Pampulha – Belo Horizonte – MG – Brazil

annaclaudiaphn@gmail.com, anisio@decom.cefetmg.br,
glpappa@dcc.ufmg.br, cardeal@decom.cefetmg.br

Abstract. *Everyday millions of images are uploaded to photo sharing social networks, such as Instagram, and recommender systems are a useful tool to guide the user to find content that interests him. This paper proposes a multi-modal recommender framework tested under the task of recommending images to these users. Traditional state-of-the-art item-based recommender systems suffer from the “curse of dimensionality” problem. For instance, in photo sharing websites, images can be described using multiple modalities (e.g., image content and textual description). The high-dimensional image features lead to a degraded recommendation performance since the distribution of the original data is extremely sparse. In order to alleviate this problem, we propose to project this multi-modal high-dimensional data into a common and lower dimensional latent factor space. As shown in our experiments, our framework provides gains from 15% to 17% in terms of hit-rate over the baselines.*

1. Introduction

On a daily basis, users upload millions of photos in image sharing social networks including Flickr, Pinterest, and Instagram. In 2016, Flickr users shared, on average, 51 million images per month¹. The availability of this massive amount of information is useless if one is unable to effectively find relevant information that matches their preferences.

Recommender systems are gaining widespread importance in Web applications as they help users to deal with the information overload problem [3]. Several recommendation algorithms have been proposed in the literature [5, 8, 17, 18], and they can be categorized as (i) latent space methods and (ii) neighborhood-based methods (user-based and item-based). These methods can be used to solve the rating prediction task, in which we want to infer the rating a user would give to an item, or the top- N recommendation task, in which we want to find a list of items ordered by their relevance to the user. Latent-space methods have been shown to be superior for solving the rating prediction task, whereas

¹<https://www.flickr.com/photos/franckmichel/6855169886> (accessed in 04/18/2017)

neighborhood methods are better at the top- N recommendation task [8, 12]. Among the latter, item-based methods – which are based on similarities between items calculated by a predefined measure – have shown to outperform the user-based approaches [8].

However, item-based methods suffer from performance problems when items are described in high-dimensional spaces, which are also very sparse. In order to alleviate the problem of high dimensionality, the authors in [4] introduced Sparse Linear Methods (SLIM). SLIM deals with high dimensional data (i.e., the curse of dimensionality) by computing an item recommendation score based on a sparse aggregation of the user items (see Section 3.2 for details). SSLIM (Side information SLIM) [17] builds up on SLIM by also taking into account side information about the items, such as the director of a movie or the genre of a song, instead of focusing only on past preferences of users per items.

This paper deals with the problem of multi-modal recommendation, which is also extremely high-dimensional and sparse. Images, which are the items recommended in this paper, are often described in terms of at least two modalities: (i) textual (e.g., description) and (ii) visual (e.g., color). Although multi-modal representations are high-dimensional and sparse, a direct application of SLIM or SSLIM would imply merging two different feature spaces (e.g., text and color) into a single one, which does not necessarily make the most of the information available.

In this direction, this paper introduces LDSSLIM (Low-Dimensional Side information SLIM). LDSSLIM is a framework developed for top- N multi-modal recommendation problems. We assume we have a history of past user preferences, and multi-modal descriptions of the images, and need to return the N most relevant new images. Since we directly represent the similarities among items in our side information matrix, our approach has the advantage of capturing the feature-based relations among items. The framework proposes the use of dimensionality reduction methods to combine multi-modal image description through a common latent space of features. This is a principled way to describe the images that also overcome the high-dimensionality problem. We emphasize that our framework is flexible to consider different dimensionality reduction methods and item similarity measures.

The key contributions of this work are threefold:

- We provide a general framework for the top- N image recommendation problem that is flexible to combine multi-modal representations.
- We propose and discuss a set of alternative instantiations for the proposed framework for image recommendation, in order to highlight the behavior and effectiveness of each possible choice.
- We advance the state-of-the-art of item-based methods by using these instantiations into the image recommendation problem.

The rest of the paper is organized as follows. In Section 2, we review the related work. In Section 3, we introduce our proposed recommendation method. In Section 4, we detail the experimental setup. In Section 5, we present and discuss the results. Finally, in Section 6, we conclude the paper and discuss possible future work.

2. Related work

Recommendation methods can be classified into two types: (i) latent space methods and (ii) neighborhood-based methods (user-based and item-based). This paper focuses on the

latter, which are the best for solving the top- N recommendation task. Neighborhood-based methods use a predefined measure of similarity to calculate how similar items or users are. In the case of item-based method, the similarity of items is calculated, hence the N most similar items to those already rated by the user recommended. In the case of user-based methods, the similarity is calculated considering the profile of users, and the items of similar users recommended as those of interest [3]. In [8], the authors presented a nearest neighbor item-based method and showed that item-based approaches lead to better top- N recommendation than user-based ones.

As previously explained, the framework proposed in this work is motivated by a class of item-based methods known as *Sparse Linear Methods* [17, 18], which were developed for top- N recommendation. The first member of the class is the **SLIM** recommender algorithm, which considers only the rating matrix as input to compute recommendations. The other important member of the sparse linear class of methods is known as **SSLIM**, which proposes to exploit, besides the rating matrix, the descriptive information about the items. This class of recommendation methods represents the state-of-the-art in top- N recommender systems. We detail both **SLIM** and **SSLIM** in Section 3.2. In [5], the authors proposed the PureSVD algorithm, which uses a truncated SVD (Singular Value Decomposition) of the ratings matrix to compute the top- N recommendations. Besides the sparse linear methods have demonstrated superior performance than previously approaches [18], we opt to also include them as our baselines.

Concerning multi-modal image recommendation, the authors in [1] presented a recommendation method based only on the descriptive features of items. Since they did not use the rating matrix, the method presents a poor performance on suggesting images. In [11], the authors presented a recommendation system for fashion related images, which suggests similar items based on clustering of images using a Gaussian Mixture Model. Different from their work, we do not restrict the system to a certain domain (e.g., fashion related images). As most recommendation problems need to deal with high dimensional data, dimensionality reduction representation of users and items have been successfully applied in recommender systems [13, 19]. In [19], the authors focus on exploiting a single modal item representation to reduce the dimensionality of item representation by using the Latent Semantic Indexing model. In [13] the authors survey a series of matrix factorization techniques to reduce the ratings matrix to a low-dimensional space. However, these methods focused on a single dimension to represent the items.

3. Generalized LDSSLIM

This section details the proposed Generalized Low-Dimensional **SSLIM** (**LDSSLIM**) framework. The framework works by moving different multi-modal high-dimensional representations of items to a low-dimensional latent space. It receives as input $|M|$ items – which in our case are images – represented by a set of k features, which may represent one or more modalities and are the side information the method works with. We assume that these items covary according to a small number of explanatory variables ($|l| < |k|$), where l refers to the set of latent factors. We call these representation as latent factors because they cannot be directly observed [6].

LDSSLIM uses dimensionality reduction methods to discover and extract these l latent factors from the high-dimensional data according to a specific criteria. By mapping

these items representation to a low-dimensional latent space, we also automatically reduce the sparsity of the representation. This new low-dimensional item representation is then used to compute the similarity between pairs of items, and these similarities given to a sparse linear model to suggest relevant items to users based on their preferences expressed in matrix R . Since we use a low-dimensional item representation to regularize the original SSLIM method, our approach is referred to as and denoted by LDSSLIM.

3.1. Problem formulation

In this section, we formalize the item recommendation problem tackled in this paper and present the notation used to detail our method. Let U be the set of users and M the set of all images (i.e., items). Let $R \subseteq U \times M$ be the rating matrix, where $m_{ij} = 1$, if image m_j is relevant to user u_i , and 0, otherwise. The length of user and item sets are denoted by $|U|$ and $|M|$, respectively.

The i^{th} row of matrix R , denoted by r_i^T , stands for the preference profile of user u_i on all images, while the j^{th} column of R stands for the profile of image m_j on all users. The descriptive information on all images is given by a $k \times |M|$ matrix B , where k is the number of features in a given modality (e.g., number of words used in the textual description). The j^{th} column of F represents the descriptive information feature vector of image m_j , denoted by b_j .

The target problem consists of identifying, based on u_i 's preferences (inferred from R) and side (descriptive) information extracted from items, the top- N items that would best match u_i 's interests. This problem can be formalized as:

Definition 3.1 (Top- N recommendation) *Given matrices R and B , and a target user $u_i \in U$, suggest a ranked list of new items in M that matches u_i 's preferences.*

3.2. Sparse linear methods

As previously explained, LDSSLIM was motivated by a class of methods that were recently developed for top- N recommendation: the SSLIM family of methods [18]. These methods incorporate descriptive item information within the Sparse Linear Method (SLIM) previously proposed in [17].

In SLIM, the recommendation score on a given item m_j of a user u_i (denoted by \hat{r}_{ij}) is computed as a sparse aggregation of the items that have been rated by u_i :

$$\hat{r}_{ij} = r_i^T w_j, \quad (1)$$

where $r_{ij} = 0$ and w_j is a size- $|M|$ sparse vector of aggregation coefficients. Hence, the model can be presented as:

$$R \sim RW, \quad (2)$$

where W is a $|M| \times |M|$ sparse matrix of aggregation coefficients, and j^{th} column is w_j as in Eq. 1, and each row r_i^T of $R \sim RW$ represents the recommendation scores on all items for user u_i . The top- N recommendation list for target user u_i is computed by sorting u_i 's non-rated images based on their scores in decreasing order according to Eq. 1 and returning the top- N items.

SLIM learns the $|M| \times |M|$ matrix W in Eq. 2 as the minimizer for the following regularized optimization problem:

$$\begin{aligned} & \underset{W}{\text{minimize}} \quad \frac{1}{2} \|R - RW\|_F^2 + \frac{\beta}{2} \|W\|_F^2 + \lambda \|W\|_1, \\ & \text{subject to } W \geq 0 \text{ and } \text{diag}(W) = 0, \end{aligned} \quad (3)$$

where $\|W\|_1$ is the element-wise ℓ_1 -norm of W and $\|\cdot\|_F$ is the matrix Frobenius norm, and β and λ are regularization parameters. The above constraints are necessary to avoid the trivial solution (i.e., W is an identity matrix) and to ensure that r_{ij} is not used to compute \hat{r}_{ij} , respectively.

Note that SLIM is a general framework that computes the aggregation coefficient matrix W using only the rating matrix R . The SSLIM family of methods, in contrast, adds descriptive item information to the problem to improve the quality of SLIM recommendations. Hence, besides the rating matrix R , SSLIM considers a descriptive item matrix B when learning W . In SSLIM, the objective is to learn W satisfying two equations – $\hat{R} \sim RW$ and $\hat{B} \sim BW$, as follows:

$$\begin{aligned} & \underset{W}{\text{minimize}} \quad \|R - RW\|_F^2 + \frac{\alpha}{2} \|B - BW\|_F^2 + \frac{\beta}{2} \|W\|_F^2 + \lambda \|W\|_1, \\ & \text{subject to } W \geq 0 \text{ and } \text{diag}(W) = 0, \end{aligned} \quad (4)$$

where α parameter balances the use of collaborative filtering (matrix R) and side-information (matrix B), and β and λ are regularization parameters.

Since the columns of W are independent, the optimization problem of both methods can be decoupled into a set of optimization problems of the form:

$$\begin{aligned} & \underset{W}{\text{minimize}} \quad \|a_i - Aw_i\|_2^2 + \frac{\alpha}{2} \|b_i - Bw_i\|_2^2 + \frac{\beta}{2} \|w_i\|_2 + \lambda \|w_i\|_1 \\ & \text{subject to } W \geq 0 \text{ and } \text{diag}(W) = 0 \end{aligned} \quad (5)$$

Both SSLIM and SLIM focus on sparse data and can be executed in parallel, which makes them particularly interesting to the domain of image recommendation, where datasets tend to be large.

3.3. From SSLIM to LDSSLIM

LDSSLIM assumes that the aggregation coefficient matrix W of SSLIM can be represented as a function of the item feature space. In other words, we represent directly the similarities among items in our side information matrix. This approach has the advantage of capturing the feature-based relations among items.

To account for that, we build an item-item similarity matrix L by exploiting the low-dimensional representation of items. It is important to note that we opt to use a low-dimensional representation of items because raw item descriptions are very sparse (e.g., bag-of-words for textual modality). Note that the sparse linear models propose to use the $l-1$ -norm to build sparse models. We also used this norm, and, additionally, emphasize deals with item representation sparsity by representing them in a lower dimensional space.

Algorithm 1 Generalized LDSSLIM

Require: Matrices R and $\{\mathbf{B}\}$, low dimensional method f , and similarity function g

```

1: for  $i = 1, \dots, |M|$  do
2:    $l_i \leftarrow \text{low\_dim}(\{b_i\})$  ▷ Low-dimensional item representation
3: end for
4: for  $i = 1, \dots, |M|$  do ▷ Computing items similarities
5:   for  $j = 1, \dots, |M|$  do
6:      $L_{ij} \leftarrow \text{sim}(l_i, l_j)$ 
7:   end for
8: end for
9: for  $i = 1, \dots, |M|$  do ▷ Computing the aggregation coefficient matrix
10:   $W \leftarrow \underset{W}{\text{argmin}} \left\| a_i - Aw_i \right\|_2^2 + \frac{\alpha}{2} \left\| l_i - Lw_i \right\|_2^2 + \frac{\beta}{2} \|w_i\|_2 + \lambda \|w_i\|_1$ 
11:  subject to  $W \geq 0$  and  $\text{diag}(W) = 0$ 
12: end for

```

The main difference between LDSSLIM and SSLIM is on how the matrix W is learned. LDSSLIM algorithm for learning matrix W is presented in Algorithm 1. Given W , we use Eq. 1 to perform recommendations in the test dataset. The algorithm receives as input the same matrices as SSLIM, i.e., the rating matrix R and the descriptive item matrix B . However, instead of using a single descriptive matrix B , we use a set of descriptive item matrices $\{\mathbf{B}\}$. Each matrix $B \in \{\mathbf{B}\}$, provides a distinct modal representation of the items. For instance, if the items are described using textual or visual information, for which case we have $\{\mathbf{B}\} = \{B_{\text{textual}}, B_{\text{visual}}\}$. LDSSLIM exploits these multimodal informative item description by representing the items in a common latent space using dimensionality reduction methods.

LDSSLIM is flexible in the sense that one can set as inputs the dimensionality reduction method f and the similarity function among items g . The algorithm works in three sequential steps. The first step generates the low-dimensional representation of the items (Lines 1–3). The only restriction here is that the length of the low-dimensional representation needs to be smaller than or equal to the original representation of item i , i.e., $|l_i| \leq |b_i|$. The second step identifies how similar are the pairs of items in terms of the given similarity function g (Lines 4–8). Finally, the last step computes the aggregation coefficient matrix W . Here, we follow the optimization scheme of SSLIM and implement an coordinate descent algorithm to find W (Lines 9–12).

The original SLIM and SSLIM methods are unable to use the improvements proposed in this work, namely: (i) multimodal item representation on a low-dimensional latent space, and (ii) explicit representation of similarities among items.

3.4. Possible Instantiations

In this section, we emphasize the flexibility of our proposed framework by presenting a set of possible instantiations for each of its components. We consider that this flexibility is a key advantage of our approach over the state-of-the-art algorithms.

We use a list of possibilities that highlight the properties of the framework to suggest relevant items to users, which includes three dimensionality reduction methods

and three different distance metrics. This is not an exhaustive list and other instantiations may be used.

Regarding dimensionality reduction methods, they can be roughly divided into two categories: (i) unsupervised and (ii) supervised. Basically, unsupervised dimensionality reduction aims to optimize the minimization of data loss information, while supervised dimensionality reduction methods aim to maximize inter-class discriminant information. Since recommender systems are by definition an unsupervised learning problem, we use here the following unsupervised dimensionality reduction methods:

- Principal component analysis (PCA) [9] performs reduction by embedding the data into a linear subspace of lower dimensionality. The main advantages of PCA are that it has an optimal reconstruction error and convenient calculation.
- Latent semantic analysis (LSA) [15] is based on singular value decomposition (SVD) to find a low-rank approximation of the original data. This low-rank matrix refers to the data represented in the low-dimensional space. The method is efficient and consistent (i.e., insensitive to starting conditions).
- Isometric Mapping (ISOMAP) [20] uses a geodesic (or curvilinear) distance induced by a neighborhood item graph. The advantage of Isomap is that it better preserves geometric structure of the data.

These methods were combined with three similarity metrics: cosine, Manhattan and Euclidean².

4. Experimental setup

In this section, we describe our experimental setup. We have made all code used in this paper available to ensure that our results are reproducible and extensible.³

4.1. Dataset

To evaluate our image recommendation method, we used a real-world image dataset that contains 1275 Flickr users, 1000 images and 1177 ratings [10]⁴. The images are quite diverse, covering topics like nature, portraits, etc. For each image, we extracted text from the image description and visual information from the image pixels. The textual information used refers to TF (term frequency) and TFIDF (term frequency x inverse document frequency), the bag-of-words (BOW) vector is 13,564-dimensional. The visual information refers to color, SIFT (1000 dimensions) [16], SURF (1000 dimensions) [2], HOG (3780 dimensions) [7].

4.2. Evaluation methodology

We employed a 5-fold cross validation procedure to evaluate the performance of the proposed method. For each user, we randomly select 20% of their evaluated images and placed in the test set. The remainder 80% of user data was placed in the training set. To validate the performance of our approaches, we use the Wilcoxon signed-rank test to assess the statistical significance of our results. We consider a 95% significance level for all

²Since these three metrics refer to distance, we use $1 - \text{distance-value}$ to compute the similarities of our side-information matrix

³Sent under request.

⁴We sampled the 1000 images with the largest number of words in their description.

comparisons. We opt to use the Wilcoxon signed-rank test since it is a non-parametric statistical hypothesis testing procedure that requires no previous knowledge of the samples distribution.

The statistical tests are computed over the values for Hit-rate (HR), and Average-Reciprocal Hit Rate (ARHR) at the top- N suggested images, which consists of the two most important and frequently used performance metrics to evaluate a given permutation of a ranked list [3]. We report the results of metrics averaged over users and test sets. The metrics are defined as follows;

$$HR@N = \frac{\#hits}{\#users} \quad (6)$$

$$ARHR@N = \frac{1}{\#users} \sum_{i=1}^{\#hits} \frac{1}{rank_i} \quad (7)$$

where $\#users$ is the total number of test users and $\#hits$ is the number of users for whom the recommender suggests relevant items in the size- N recommendation list and $rank_i$ is the position of the test item in the top- N recommendation list for the i^{th} hit. Note that ARHR is a weighted version of HR, since it measures the inverse of the position of the suggested item in the ranked list.

Using these metrics, LDSSLIM is compared to two state-of-the-art methods for top- N recommendation that serve as inspiration to it, namely, SLIM and SSLIM. Besides the aforementioned sparse linear methods (i.e., SLIM and SSLIM), we also compare our approach to the following methods: (i) PureSVD, which factorizes the rating matrix R by using a factorization technique similar to the Singular Value Decomposition method, and (ii) WeightedItemKNN, which is a recommendation algorithm representative of neighborhood methods. The parameters of PureSVD refers to the number of latent factors, and the parameters of WeightedItemKNN refers to the number of neighbors used to predict the recommendation.

5. Results and discussion

In this section, we empirically validate our LDSSLIM framework for multi-modal image recommendation. In particular, the research questions guiding the remainder of the paper are:

Q1: Do similarity models based on dimensionality reduction improve the state-of-the-art recommendation methods?

Q2: What is the effect of dimensionality reduction methods on the recommendation accuracy?

Q3: What is the effect of varying the number of latent factors for image representation on the recommendation accuracy?

Q4: What is the effect of using different similarity metrics on the recommendation accuracy?

Table 1. Performance obtained by image recommendation methods considering $N = 1$. The gains of our proposed framework are relative to the best baseline method and are presented in parentheses. The best baseline performance is represented in *italic*.

Method	Params	HR@1 (gain)	Params	ARHR@1 (gain)	
Baselines	WeightedItemKNN	100	150	0.0062	
	PureSVD	50	50	0.0067	
	SLIM	-	-	0.0080	
	SSLIM-textual	TFIDF:Euc	<i>0.0314</i>	TFIDF:Euc.	0.0128
	SSLIM-visual	Hog:Euc	0.0011	Surf:Cos	0.0020
	SSLIM-concat.	TF+Surf:Euc	0.0320	TF+Surf:Euc	0.0131
LDSSLIM	PCA:400 TF+HoG:Euc	0.0369 (0.1531)	LSA:300 TFIDF+Sift:Euc	0.0154 (0.1755)	

Q5: What is the effect on the recommendation accuracy of varying the number of suggested images to users?

In Table 1, we list the scores of the state-of-the-art baseline methods and LDSSLIM in terms of HR@1 and ARHR@1. For the baselines, the results reported encompass only the best combination of metric and image representation for each modality. Note that the performance of PureSVD and WeightedItemKNN is worst than both the sparse linear methods and our proposed framework. We also present the results of SSLIM when concatenating all available information into a single matrix (SSLIM-concat). We considered only the first position of the recommended list (i.e., $N = 1$) because most of our users evaluated few images. For all metrics, the proposed framework outperforms the baselines with gains of 15% and 17% in terms of HR and ARHR metrics, respectively. Note that the results of SLIM are really low as it does not consider side information, and our rating matrix is extremely sparse. The best results of HR@1 were obtained using only textual features, while the best results of ARHR@1 were obtained when using both descriptors. From these results, we can conclude that considering a low-dimensional representation of multi-modal information improves the quality of recommendation accuracy.

But what are the impacts of the parameters of the frameworks to the results? We start by looking at the dimensionality reduction method and its impact on the recommendation. As we can see in Figure 1, the LSA reduction method achieves the best results over the best baseline. As we are interested in suggesting a ranked list of N relevant images to users, Figure 1 also varies N . As expected, the performance of all recommenders tend to increase as we increase the number of recommended images. This happens because the algorithm is more likely to recommend relevant images when we increase the number of total suggested images.

Next, we zoom in on the impact of the dimensionality reduction over our method. Figure 2 shows how the number of latent factors impacts the performance of the recommendation. The results show that the proposed framework achieves the best results when considering more than 300 latent factors to represent images.

The second parameter of LDSSLIM is the similarity function, which is used to

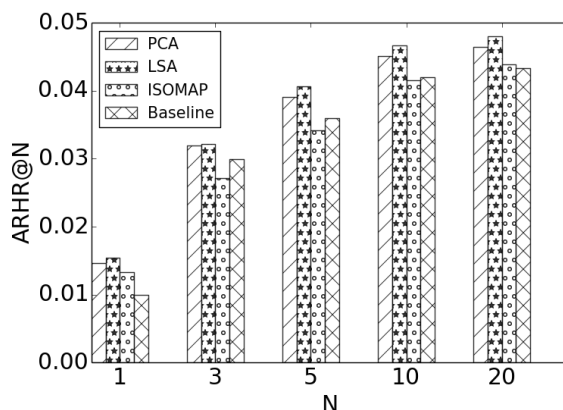


Figure 1. Varying the length of the top- N list.

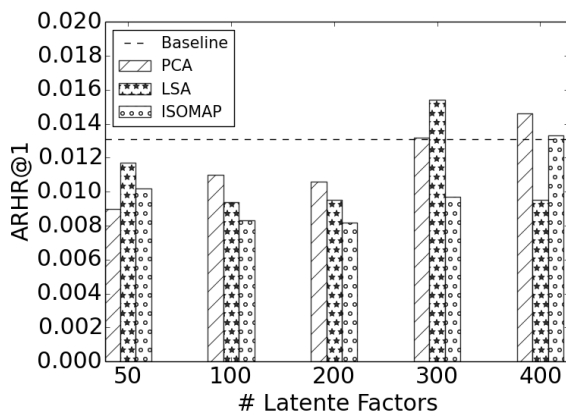


Figure 2. Varying the number of latent factors.

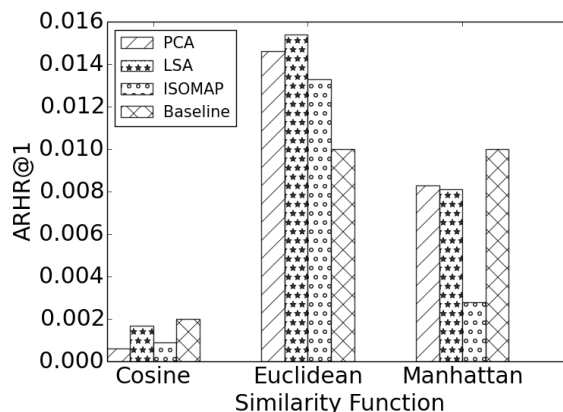


Figure 3. Varying the similarity measure.

compute how related two items are. In Figure 3, we present the performance of the image recommenders in terms of ARHR@1 the set of distinct similarity functions. We can see that the similarity function affects the performance of each method, including our proposed framework and the baseline. Overall, best results were obtained with the Euclidean distance.

In order to illustrate the types of results returned by the method, Figure 4 shows an example of image recommended to a user given its already rated images. The three images from these user are somehow related to art (paintings and stained glass), and the suggested image brings a sculpture, which we consider a good recommendation.

6. Conclusion and future work

In this work, we propose a generalized framework for image recommendation called LDSSLIM, which leverages low-dimensional image representation to suggest relevant recommendations to users. The low-dimensional representation of images overcomes

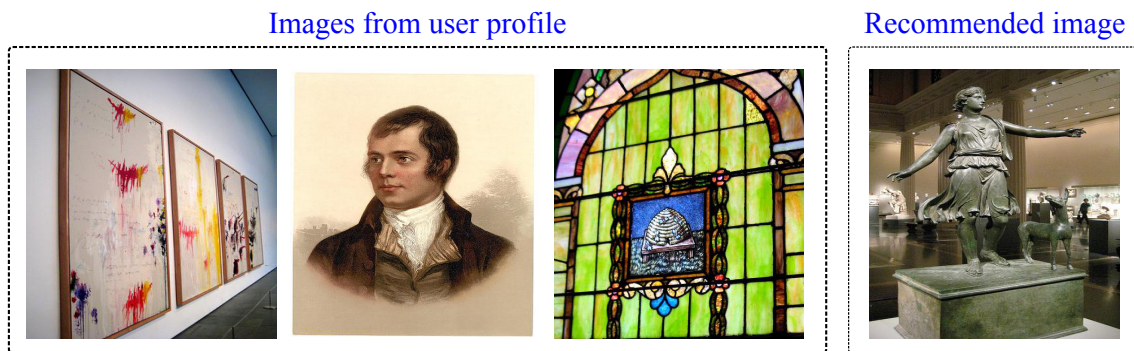


Figure 4. Illustrative example of an image recommendation performed by our framework.

the curse of dimensionality problem, which is a drawback in item-based recommender systems. Several instantiations of our proposed framework were tested by combining different parameters, namely, (i) raw textual image representation, (ii) raw visual image representation, (iii) dimensionality reduction method, (iv) number of latent factors, and (v) similarity function. As our results show, our proposals presented gains over the state-of-the-art item-based recommender algorithms. Although the list of investigated instantiations are representative, the space of possible choices is rather large. Hence, there is other possibilities for future investigations. For instance, a possible representation of visual features by using Convolutional Neural Networks [14] is being investigated. Another line of investigation refers to properly building the similarity matrix by eliminating spurious similarities.

Acknowledgements

The authors gratefully acknowledge the financial support of CEFETMG; CNPq under Proc. 431458/2016-2; and FAPEMIG under Procs. Universal FAPEMIG, process number APQ-03445-16, and FAPEMIG-PRONEX-MASWeb, Models, Algorithms and Systems for the Web, process number APQ-01400-14.

References

- [1] Marta Barrilero, Silvia Uribe, Maria Alduan, Faustino Sánchez, and Federico Alvarez. In-network content based image recommendation system for content-aware networks. In *IEEE Conf. on Computer Communications*, pages 115–120, 2011.
- [2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *ECCV 2006*, pages 404–417, 2006.
- [3] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.
- [4] Evangelia Christakopoulou and George Karypis. Local item-item models for top-n recommendation. In *ACM RecSys*.
- [5] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the 4th ACM conference on Recommender systems*, pages 39–46, 2010.

- [6] John P Cunningham and M Yu Byron. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509, 2014.
- [7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR'05*, volume 1, pages 886–893, 2005.
- [8] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.
- [9] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- [10] Mark J. Huiskes and Michael S. Lew. The MIR Flickr Retrieval Evaluation. In *ACM International Conf. on Multimedia Information Retrieval*, pages 39–43, 2008.
- [11] Vignesh Jagadeesh, Robinson Piramuthu, Anurag Bhardwaj, Wei Di, and Neel Sundaresan. Large scale visual recommendations from street fashion images. In *ACM SIGKDD*, pages 1925–1934, 2014.
- [12] Santosh Kabbur, Xia Ning, and George Karypis. Fism: factored item similarity models for top-n recommender systems. In *ACM SIGKDD*, pages 659–667, 2013.
- [13] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.
- [14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [15] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.
- [16] David G Lowe. Object recognition from local scale-invariant features. In *IEEE international conference on Computer vision*, volume 2, pages 1150–1157, 1999.
- [17] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In *ICDM'11*, pages 497–506, 2011.
- [18] Xia Ning and George Karypis. Sparse linear methods with side information for top-n recommendations. In *ACM RecSys*, pages 155–162, 2012.
- [19] Panagiotis Symeonidis. Content-based dimensionality reduction for recommender systems. In *Data Analysis, Machine Learning and Applications*, pages 619–626. Springer, 2008.
- [20] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Aplicação de Redes Neurais Convolucionais na Identificação de Emoções em Textos

Zacarias Curi¹, Alceu de Souza Britto Jr¹, Emerson Cabrera Paraiso¹

¹Laboratório de Sistemas Inteligentes - LASIN
Programa de Pós-Graduação em Informática – PPGIA
Pontifícia Universidade Católica do Paraná - PUCPR - Curitiba, PR – Brasil

{zacarias, alceu, paraiso}@ppgia.pucpr.br

Abstract. *Identifying emotions is one of the most popular tasks in the natural language processing area. The use of deep learning techniques has brought excellent results in this area. After a bibliographic review, we did not find articles that use deep learning techniques to identify emotions in texts written in Portuguese. In order to verify the possibility of using and the efficiency of these techniques in this type of problem, we propose the application of the Convolutional Neural Networks (CNNs) to the classification of a corpus annotated with the Ekman's six basic emotions and polarity. The method proved to be equivalent to the Support Vector Machines (SVM) for the data labeled with Ekman's emotions, although for the annotations with polarity, the proposed method outperformed the SVM.*

Resumo. *A identificação de emoções é uma das tarefas mais populares da área de processamento de linguagem natural. O uso de técnicas de Aprendizagem Profunda tem proporcionado excelentes resultados nessa área. Contudo, após realização de um levantamento bibliográfico não foram encontrados artigos utilizando Aprendizagem Profunda para identificar emoções em textos escritos em português. Buscando verificar a possibilidade de utilização e a eficiência dessas técnicas nesse tipo de problema, propomos a aplicação das CNNs para a classificação de um corpus anotado com as seis emoções básicas de Ekman e com polaridade. O método se apresentou equivalente ao SVM para a base rotulada com emoções e apresentou um resultado superior ao SVM para a base rotulada com polaridade.*

1. Introdução

O processamento de linguagem natural (PLN) abrange um conjunto de técnicas para analisar computacionalmente textos escritos em linguagem natural. A sua finalidade é aproximar-se da capacidade humana em diversas tarefas e aplicações [Liddy 2001]. Atualmente, a identificação de emoções em textos é uma das tarefas mais populares do PLN.

A tarefa de identificação de emoções teve a sua popularidade ampliada com o desenvolvimento da Web 2.0. Diferentemente dos Web sites onde as pessoas acessam a informação de forma passiva, a Web 2.0 permite que os usuários manifestem suas opiniões sobre o conteúdo. As notícias disponibilizadas em sites online também podem ser analisadas, com o objetivo de identificar os aspectos emocionais nelas presentes.

Esse tipo de texto apresenta algumas características que o torna diferente de textos extraídos de outras fontes, como redes sociais. Sua linguagem é formal e suas emoções geralmente são implícitas. Outra característica desse tipo de texto foi observada por [Dosciatti et al. 2015], que mostraram que as notícias com polaridade negativa (associadas a emoções negativas, como tristeza ou raiva) ocorrem com mais frequência que as notícias positivas.

A maior parte dos trabalhos relacionados a identificação de emoções utilizam apenas a polaridade dos textos, onde existem apenas duas categorias, positivo e negativo [Aman e Szpakowicz 2007]. Outras abordagens menos utilizadas realizam a identificação através de categorias de emoções como o modelo proposto por [Ekman 1992], utilizado por [Alm et al. 2005], [Strapparava e Mihalcea 2008], [Chaffar e Inkpen 2011], [Ferreira et al. 2015] e [Volkova et al. 2015].

Os métodos para a identificação de emoções podem ser classificados em supervisionados, não supervisionados e semi supervisionados. Atualmente uma vasta quantidade de técnicas supervisionadas para realizar a análise estão presentes na literatura, como as vistas em [Vinodhini e Chandrasekaran 2012]. Novas técnicas, como as relacionadas à Aprendizagem Profunda, vêm se destacando na área e gerando melhores resultados em diversas bases [LeCun et al. 2015].

Segundo [LeCun et al. 2015], a Aprendizagem Profunda é uma abordagem que permitiu grandes avanços na resolução de problemas que desafiaram a área de Inteligência Artificial por vários anos. Como visto por [Tang et al. 2015] a utilização de Aprendizagem Profunda vem se destacando na área de PLN como um todo, incluindo a identificação de emoções em textos.

Após uma pesquisa bibliográfica não foram encontrados artigos utilizando técnicas de Aprendizagem Profunda para identificar as emoções básicas, propostas por [Ekman 1992], em textos escritos em português. Para verificar a possibilidade de utilização e a eficiência da técnica em múltiplas categorias de emoção, propomos a utilização das Redes Neurais Convolucionais (CNN do inglês *Convolutional Neural Network*) na identificação de emoções em textos. Os resultados obtidos demonstram que a CNN se apresentou tão eficiente quanto o SVM para a base rotulada com seis emoções e superior ao SVM para a base rotulada com polaridade. Os diferentes resultados para as bases pode ser justificado pela pequena quantidade de exemplos por classe na base com seis emoções, causando *overfitting* na rede.

Este artigo está organizado em 6 seções. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta o modelo de CNN utilizado neste trabalho. A Seção 4 apresenta os experimentos realizados, assim como as configurações da rede utilizada, enquanto a Seção 5 apresenta os resultados do experimento realizado. A conclusão e os trabalhos futuros são apresentados na Seção 6.

2. Trabalhos Relacionados

Diversos trabalhos existentes na literatura apresentam técnicas para identificação de emoções em textos utilizando algoritmos de Aprendizagem Profunda. Embora nenhum trabalho tenha sido encontrado utilizando as seis emoções básicas propostas por Ekman para o português outras categorias de emoções e idiomas foram explorados. Em

[Li et al. 2016] os autores apresentam um método utilizando *Restricted Boltzmann Machine* (RBM) para classificar duas bases rotuladas com categorias de emoções. Uma das bases utilizadas consiste de 1.246 títulos de notícias escritas em inglês e anotadas com as 6 emoções propostas por Ekman. Li e seus colegas também utilizaram uma base com 7.666 sentenças anotadas em alegria, culpa, desgosto, medo, raiva, tristeza e vergonha.

Outro trabalho utilizando Aprendizagem Profunda para classificação de categorias de emoções em textos foi apresentado por [Tang et al. 2013], onde o algoritmo *Deep Belief Network* (DBN) é utilizado para classificar um corpus anotado com os rótulos: alegria, raiva, surpresa e tristeza. No artigo, Tang e colegas apresentam o processo de montagem do corpus, onde textos em chinês foram automaticamente extraídos do *microblog sina* utilizando *emoticons* referentes a cada emoção utilizada. Embora a maioria dos trabalhos utilizando Aprendizagem Profunda para a identificação de emoções em textos utilize os idiomas inglês e chinês, o trabalho de [Brum et al. 2016] apresenta o uso do algoritmo *Recursive Neural Tensor Network* (RNTN) para classificação de 12.508 sentenças extraídas de um site de *reviews* de livros em português rotuladas em polaridade (positivo, negativo) e polaridade + neutro.

Além da polaridade e classes de emoções outros tipos de rótulos foram utilizados com algoritmos de Aprendizagem Profunda, inclusive com a CNN. Um exemplo disso é o trabalho de [Kim 2014], onde essa rede foi utilizada com sete diferentes corpora, contendo de 2 a 6 classes. Uma das bases utilizadas pelo trabalho de Kim é composta por 10.662 *reviews* de filmes escritos em inglês e anotados com polaridade. A mesma base foi utilizada por [Ouyang et al. 2015], onde os autores utilizam uma combinação do algoritmo *word2vec* e CNN para realizar a classificação. Outro trabalho apresentando uma combinação de uma técnica de *embedding* com a CNN para a classificação de polaridade em *reviews* foi apresentado por [Johnson e Zhang 2015], onde um método semi supervisionado é utilizado. O trabalho de [Severyn e Moschitti 2015] também apresenta um método utilizando CNN para classificação de polaridade, mas o autor utiliza uma base composta de *tweets* para validar seu método. *Tweets* também foram utilizados por [Stojanovski et al. 2015], onde a CNN foi utilizada para classificar as seis emoções propostas por [Shaver et al. 1987] (amor, alegria, surpresa, raiva, tristeza e medo) em mensagens referentes a copa do mundo da FIFA.

No trabalho de [Jurgovsky e Granitzer 2015] as arquiteturas CNN e *Semi-Supervised Recursive Autoencoder* (SRAE) foram comparadas na tarefa de classificação de polaridade. Os autores concluíram que uma CNN, mesmo com apenas uma camada de convolução, apresenta um resultado superior ao SRAE. Além do SRAE, o resultado superior a diversos métodos clássicos, como SVM, Naive Bayes e C4.5, foi demonstrado em diversos dos artigos citados anteriormente.

A CNN foi criada para a utilização em imagens [LeCun et al. 1998]. Os trabalhos utilizando CNN citados anteriormente apresentam algumas alterações que permitem a utilização desse algoritmo com informações textuais. Na seção a seguir é apresentado o funcionamento dessa arquitetura e como ela pode ser utilizada para a tarefa de identificação de emoções em textos.

3. Redes Neurais Convolucionais na Identificação de Emoções em Textos

Segundo o trabalho realizado por [Schmidhuber 2015], a principal inspiração para a criação de algoritmos de Aprendizagem Profunda foi o princípio apresentado por [Hubel e Wiesel 1959] e [Hubel e Wiesel 1962], onde o córtex visual de gatos foi estudado. Nos trabalhos de Hubel e Wiesel foi identificado que as células do córtex são ativadas com pequenas características visuais, como a orientação de linhas, que se comunicam entre si para reconhecer os padrões existentes nos objetos.

O avanço nos estudos do funcionamento dos neurônios e os estudos de redes neurais com múltiplas camadas por [Ivakhnenko e Lapa 1966] permitiram que pesquisadores utilizassem novas formas de representação dos dados. [Fukushima e Miyake 1980] apresentam um método capaz de dividir o problema original da mesma forma que o cérebro de animais. Atualmente esse método é chamado de Rede Neural Convolutiva e diversos trabalhos utilizam uma versão melhorada dessa rede. As principais modificações realizadas com o passar do tempo foram o ajuste dos pesos com o algoritmo *backpropagation* por [LeCun et al. 1990] e o *downsampling* com o algoritmo *max pooling*.

Uma ilustração do funcionamento dos modelos atuais da CNN é apresentada na Figura 1. Como visto no modelo apresentado por [Krizhevsky et al. 2012], as CNNs são redes do tipo *feed-forward* compostas por uma ou mais camadas de convolução. Cada camada possui um conjunto de máscaras que após serem aplicadas têm seus resultados combinados. Outra função da convolução é criar subamostras da imagem original, permitindo a criação de imagens menores, com resultados diferentes. Após o fim das etapas de convolução a operação de *max pooling* é realizada sobre os mapas de características para se obter o maior valor existente em cada mapa, com isso as características mais importantes são extraídas. Por fim as características obtidas na função de *max pooling* são utilizadas como entrada em uma única camada cuja saída é a probabilidade de cada rótulo ser verdadeiro.

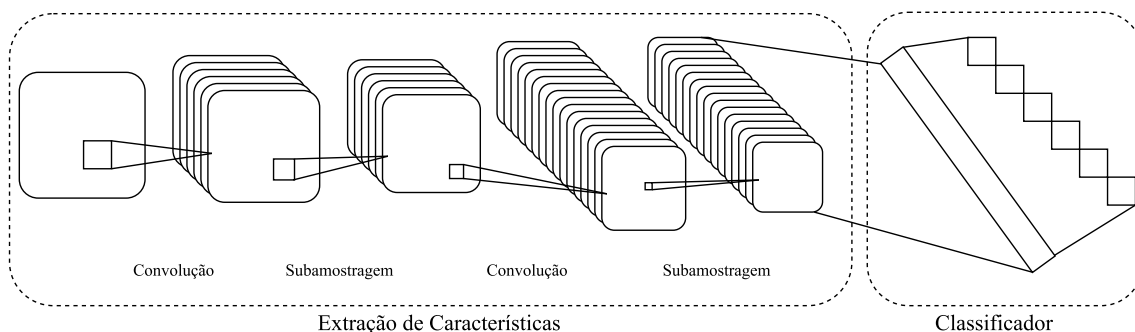


Figura 1. Funcionamento de uma CNN

A principal vantagem do processo realizado pela CNN, nas aplicações em textos, é a extração automática de características realizada pelas camadas de convolução. Originalmente a CNN foi desenvolvida para a classificação de imagens, mas com o passar do tempo diversos trabalhos apresentaram alterações na estrutura dos dados para possibilitar a utilização de áudios e textos [LeCun et al. 2015]. Uma série de ferramentas surgiu para auxiliar desenvolvedores a utilizar algoritmos de Aprendizagem Profunda, como a CNN, tanto comercialmente quanto no âmbito da pesquisa. Entre as principais ferra-

mentas existentes estão: *caffe*¹, *Deeplearning4j*², *theano*³, *TensorFlow*⁴, *keras*⁵ e *Torch*⁶. Independente da ferramenta escolhida, para a utilização da CNN na tarefa de identificação de emoções em textos, é necessário realizar uma transformação chamada de *embedding*, onde palavras são transformadas em números reais e adicionadas a um vetor, como no método apresentado em [Mikolov et al. 2013]. Outra modificação para a utilização em dados textuais é a alteração do tamanho das máscaras na operação de convolução, onde se deve considerar uma operação linear. Essas adaptações foram propostas inicialmente no trabalho de [Collobert et al. 2011]. Para possibilitar melhores resultados também é possível realizar um pré processamento dos dados em uma etapa anterior as adaptações propostas por Collobert e colegas.

A operação de pré-processamento consiste na realização de modificações nos textos para possibilitar, principalmente, uma redução na quantidade de palavras utilizadas para a classificação. Como apresentado por [Bakliwal et al. 2012], uma dessas estratégias é a substituição de símbolos por *tokens* com o seu significado, como em alguns casos onde números e símbolos monetários são substituídos por uma palavra para os referenciar. O objetivo dessa operação é representar de forma genérica algumas expressões, evitando assim instâncias pouco usadas. Outra operação com o mesmo objetivo é a aplicação de um lematizador. Essa técnica visa extrair a raiz da palavra, ignorando o tempo verbal, o gênero, o plural, entre outros. Em alguns casos, como em redes sociais, é necessário utilizar alguma técnica para remover letras repetidas, substituindo palavras escritas de maneira incorreta por sua forma original. Outra operação importante para possibilitar uma maior eficiência na classificação é a remoção dos *stop words*, onde todas as palavras que não apresentarem importância para a classificação são removidas. Adicionalmente, é possível remover as palavras menos frequentes, diminuindo a quantidade de atributos utilizados na etapa de classificação.

4. Experimentos

Nesta seção são apresentadas as configurações utilizadas e as informações sobre os experimentos realizados. Neste artigo utilizamos uma CNN baseada no método de [Collobert et al. 2011]. Nesse trabalho Collobert e colegas apresentam uma arquitetura CNN e demonstram a sua aplicação para diversas tarefas da área de PLN.

4.1. Base Utilizada

O corpus utilizado neste trabalho foi inicialmente apresentado por [Ferreira et al. 2015]. O corpus consiste de 1.312 manchetes e linhas finas (complementos das manchetes) de notícias, de diversas categorias, extraídas de sites de notícias do Brasil. A anotação do corpus foi realizada manualmente utilizando as seis emoções propostas por [Ekman 1992]. A distribuição do corpus está realizada da seguinte forma: 280 (21,34%) notícias anotadas como alegria, 226 (17,23%) como desgosto, 160 (12,2%) como medo, 168 (12,80%) como raiva, 172 (13,11%) como surpresa e 306 (23,32%) como tristeza. As emoções

¹<http://caffe.berkeleyvision.org/>

²<https://deeplearning4j.org/>

³<http://deeplearning.net/software/theano/>

⁴<https://www.tensorflow.org/>

⁵<https://keras.io/>

⁶<http://torch.ch/>

propostas por Ekman também podem ser divididas em duas categorias, positivo (alegria) e negativo (desgosto, medo, raiva e tristeza), chamadas de polaridade. O corpus anotado com polaridade contém 280 (24,56%) textos positivos e 860 (75,44%) textos negativos.

O corpus utilizado por [Ferreira et al. 2015] e neste trabalho é uma parte do corpus com 2.000 textos apresentado por [Dosciatti et al. 2015]. Segundo Dosciatti e colegas, os textos utilizados nesse corpus possuem algumas características que os tornam diferentes dos textos comumente utilizados para a identificação de emoções, como os extraídos de redes sociais. As notícias tendem a ser textos formais, com emoções implícitas, o que dificulta o processo de anotação e classificação. Para a anotação do corpus Dosciatti e colegas contaram com o apoio de cinco anotadores voluntários, todos professores do ensino superior, com experiência mínima de 15 anos em linguística. Cada texto foi anotado por dois anotadores e em casos de divergência por um terceiro anotador. O fato das emoções estarem implícitas nesse tipo de texto causou um baixo grau de concordância entre os anotadores, gerando um kappa de 0,38.

4.2. Pré processamento

Antes da realização da operação de *embedding* uma operação de pré-processamento foi realizada para reduzir o número de palavras do corpus e evitar que palavras semelhantes fossem utilizadas como palavras diferentes. Para isso, as seguintes operações foram realizadas:

- Conversão para minúsculo;
- Remoção dos *StopWords* com a lista disponibilizada pela ferramenta Snowball⁷;
- Remoção de acentos e caracteres especiais;
- Tokenização de símbolos com a substituição de números por `_num`, % por `_porc` e \$ por `_mon`.
- Aplicação do lematizador SnowballStemmer⁷ para português.

A realização dessas etapas possibilitou uma redução de 8.618 palavras para 4.169. Após a realização do pré-processamento, uma operação de *embedding* foi efetuada com a utilização da função Embedding da biblioteca keras⁸. Nesta operação todas as palavras foram mapeadas em um vetor com 1.200 números reais escolhidos aleatoriamente pela biblioteca, como apresentado no trabalho de [Gal e Ghahramani 2016]. A utilização de 1.200 posições foi realizada por ser superior ao tamanho da maior instância da base utilizada. Com a matriz montada os vetores foram utilizados como entrada da CNN. Para possibilitar uma comparação mais precisa com os resultados obtidos por [Ferreira et al. 2015] foi utilizado Validação Cruzada com 10 *folds*.

4.3. Configuração da Rede

Um algoritmo foi implementado utilizando a biblioteca keras⁸ em conjunto com a biblioteca theano⁹. Diversos testes empíricos foram realizados para verificar as melhores configurações da rede e evitar a ocorrência de *overfitting* causada pelo tamanho da base. Para a etapa de convolução da base com 6 emoções foram utilizados 320 máscaras de 4 palavras. Durante a etapa de treinamento utilizamos 300 camadas ocultas e 50 épocas.

⁷<http://snowball.tartarus.org/>

⁸<https://keras.io/>

⁹<http://deeplearning.net/software/theano/>

Na etapa de convolução da base rotulada com polaridade foram utilizados 120 máscaras de 4 palavras. Para o treinamento dessa base foram utilizados 150 camadas ocultas e 75 épocas. Os resultados obtidos e uma discussão são apresentados na seção a seguir.

5. Resultados

Após a execução dos experimentos apresentados na seção anterior obtivemos os resultados apresentados na Tabela 1 para a base com seis emoções. Para possibilitar uma comparação com os resultados existentes na literatura a tabela também apresenta os resultados obtidos por [Ferreira et al. 2015]. No trabalho de Ferreira e colegas é apresentado um método utilizando algoritmo genético para tratar o desbalanceamento da base. Como o nosso objetivo é comparar o desempenho dos algoritmos serão utilizados os resultados sem esse tratamento.

Tabela 1. Resultados obtidos para o corpus rotulado com seis emoções

Emoção	SVM [Ferreira et al. 2015]			CNN		
	Precisão	Revocação	F1	Precisão	Revocação	F1
Alegria	0,461	0,868	0,602	0,470	0,571	0,509
Desgosto	0,623	0,212	0,317	0,407	0,376	0,385
Medo	0,857	0,188	0,308	0,477	0,319	0,380
Raiva	0,764	0,327	0,458	0,529	0,375	0,435
Surpresa	0,831	0,285	0,424	0,549	0,401	0,456
Tristeza	0,428	0,758	0,547	0,469	0,607	0,525
Acurácia SVM: 50,07%			Acurácia CNN: 46,80%			

Como é possível observar na Tabela 1, a CNN utilizada neste trabalho apresentou uma acurácia inferior a obtida pelo SVM. Embora esse resultado se apresente menor é interessante avaliar os valores obtidos pelo F1 para realizar uma análise considerando o desbalanceamento dos dados. A escolha do F1 para bases de dados desbalanceadas é sugerida em trabalhos da literatura, como justificado em [Mani e Zhang 2003]. Calculando os valores do macro-F1 obtemos o valor de 44,27% para o SVM e 44,83% para a CNN. Mesmo que o SVM tenha apresentado um resultado superior à CNN em relação a acurácia, os testes estatístico de *Wilcoxon* e *teste-t* não apresentaram diferenças significativas entre os resultados do F1. Esses resultados demonstram que o algoritmo CNN se apresentou tão robusto quanto o SVM em relação a bases desbalanceadas sem a utilização de tratamentos adicionais.

A Tabela 2 apresenta os resultados obtidos com o algoritmo SVM por Ferreira e colegas, assim como os nossos resultados com a CNN. Como é possível observar o resultado obtido com a CNN apresentou um aumento de 3,68 p.p. em relação a acurácia e apresentou um F1 superior nas duas classes. O fato da CNN ter obtido um desempenho superior em relação a polaridade e inferior em relação a base com seis classes pode ser justificado pela diferente quantidade de exemplos por classe, causando *overfitting* da rede treinada. A classe negativo possui uma maior quantidade de exemplos por se tratar de um conjunto com quatro classes da base com seis emoções. Esse fato faz com que a classificação de polaridade testada se aproxime da situação ideal do algoritmo CNN, que obtém um melhor desempenho com muitos exemplos. Embora a solução ideal seja

rotular mais textos, ampliando a quantidade de exemplos na base de treinamento, algumas soluções podem ser utilizadas para otimizar o processo de treinamento com uma base pequena. Alguns trabalhos da literatura apresentam relatos em relação a quantidade de exemplos para a utilização da CNN com imagens, como em [Ng et al. 2015], onde os autores apresentaram um método utilizando *Transfer Learning* para a classificação de emoções em imagens de bases de dados pequenas. Outra abordagem para tratar esse problema foi apresentada por [Agrawal et al. 2014], onde os autores propõem o uso de uma CNN pré treinada para a utilização de uma base com apenas 5 mil imagens para treinamento.

Tabela 2. Resultados obtidos para o corpus rotulado com polaridade

Polaridade	SVM [Ferreira et al. 2015]			CNN		
	Precisão	Revocação	F1	Precisão	Revocação	F1
Positivo	0,933	0,050	0,095	0,660	0,396	0,488
Negativo	0,764	0,999	0,865	0,827	0,935	0,877
Acurácia SVM: 76,58%			Acurácia CNN: 80,26%			

6. Conclusão

Neste trabalho foi demonstrado a utilização de uma técnica com CNN para a identificação das seis emoções de [Ekman 1992] e polaridade para textos de um corpus contendo 1.312 notícias em português do Brasil. O método apresentou um resultado estatisticamente equivalente em relação ao F1 e 3,27 p.p. inferior ao SVM em relação a acurácia, para a base com as seis emoções básicas de Ekman. Para a base rotulada com polaridade o método apresentou um resultado superior em 3,68 p.p. para a acurácia e 20,3 p.p. em relação ao macro-F1. Esses resultados demonstram que a CNN se apresentou tão robusta quanto o SVM em relação a bases desbalanceadas sem a realização de tratamento para desbalanceamento. O fato da CNN ter obtido um desempenho superior em relação a polaridade e equivalente em relação a base rotulada com seis emoções pode ser explicado pela pequena quantidade de exemplos por classe, o que acaba gerando *overfitting*, como em outros trabalhos encontrados na literatura para a classificação de imagens.

Como trabalhos futuros pretendemos utilizar outros corpora com diferentes tamanhos e quantidades de classes para verificar o tamanho ideal de exemplos por classe para o uso com a CNN. Assim como diferentes corporas, pretendemos utilizar diferentes métodos de embedding para verificar qual o mais adequado para o uso em textos em português. Também temos interesse em estudar a aplicação de técnicas de *Transfer Learning* para a utilização de CNN em um corpus de textos com poucos exemplos.

Agradecimentos

Os autores agradecem a empresa Siemens Ltda que através do projeto Fábrica de Software financiou parcialmente este projeto.

Referências

Agrawal, P., Girshick, R., and Malik, J. (2014). Analyzing the performance of multilayer neural networks for object recognition. In *European conference on computer vision*, pages 329–344. Springer.

- Alm, C. O., Roth, D., and Sproat, R. (2005). Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 579–586. Association for Computational Linguistics.
- Aman, S. and Szpakowicz, S. (2007). Identifying expressions of emotion in text. In *International Conference on Text, Speech and Dialogue*, pages 196–205. Springer.
- Bakliwal, A., Arora, P., Madhappan, S., Kapre, N., Singh, M., and Varma, V. (2012). Mining sentiments from tweets. *Proceedings of the Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, 12.
- Brum, H., Araujo, F., and Kepler, F. (2016). Sentiment analysis for brazilian portuguese over a skewed class corpora. In *International Conference on Computational Processing of the Portuguese Language*, pages 134–138. Springer.
- Chaffar, S. and Inkpen, D. (2011). Using a heterogeneous dataset for emotion analysis in text. In *Canadian Conference on Artificial Intelligence*, pages 62–67. Springer.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.
- Dosciatti, M. M., Ferreira, L. P. C., and Paraiso, E. C. (2015). Anotando um corpus de notícias para a análise de sentimento: um relato de experiência. In *Proceedings of Symposium in Information and Human Language Technology*, pages 121–130. Sociedade Brasileira de Computação.
- Ekman, P. (1992). An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.
- Ferreira, L. C., Dosciatti, M. M., Nievola, J. C., and Paraiso, E. C. (2015). Using a genetic algorithm approach to study the impact of imbalanced corpora in sentiment analysis. In *Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference*, pages 163–168.
- Fukushima, K. and Miyake, S. (1980). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer.
- Gal, Y. and Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.
- Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591.
- Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154.
- Ivakhnenko, A. G. and Lapa, V. G. (1966). Cybernetic predicting devices. Technical report, CCM Information Corporation.
- Johnson, R. and Zhang, T. (2015). Semi-supervised convolutional neural networks for text categorization via region embedding. In *Advances in neural information processing systems*, pages 919–927.

- Jurgovsky, J. and Granitzer, M. (2015). Comparing recursive autoencoder and convolutional network for phrase-level sentiment polarity classification. In *International Conference on Applications of Natural Language to Information Systems*, pages 160–166. Springer.
- Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R., Hubbard, W. E., and Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, pages 396–404.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Li, X., Pang, J., Mo, B., Rao, Y., and Wang, F. L. (2016). Deep neural network for short-text sentiment classification. In *International Conference on Database Systems for Advanced Applications*, pages 168–175. Springer.
- Liddy, E. D. (2001). Natural language processing. *Encyclopedia of Library and Information Science*, 2.
- Mani, I. and Zhang, I. (2003). knn approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*, volume 126.
- Mikolov, T., Yih, W.-t., and Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.
- Ng, H. W., Nguyen, V. D., Vonikakis, V., and Winkler, S. (2015). Deep learning for emotion recognition on small datasets using transfer learning. In *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pages 443–449. ACM.
- Ouyang, X., Zhou, P., Li, C. H., and Liu, L. (2015). Sentiment analysis using convolutional neural network. In *International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 2359–2364. IEEE.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.
- Severyn, A. and Moschitti, A. (2015). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962. ACM.

- Shaver, P., Schwartz, J., Kirson, D., and O'connor, C. (1987). Emotion knowledge: further exploration of a prototype approach. *Journal of personality and social psychology*, 52(6):1061.
- Stojanovski, D., Strezoski, G., Madjarov, G., and Dimitrovski, I. (2015). Emotion identification in fifa world cup tweets using convolutional neural network. In *International Conference on Innovations in Information Technology*, pages 52–57. IEEE.
- Strapparava, C. and Mihalcea, R. (2008). Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1556–1560. ACM.
- Tang, D., Qin, B., and Liu, T. (2015). Deep learning for sentiment analysis: successful approaches and future challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(6):292–303.
- Tang, D., Qin, B., Liu, T., and Li, Z. (2013). Learning sentence representation for emotion classification on microblogs. In *Natural Language Processing and Chinese Computing*, pages 212–223. Springer.
- Vinodhini, G. and Chandrasekaran, R. (2012). Sentiment analysis and opinion mining: a survey. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(6):282–292.
- Volkova, S., Bachrach, Y., Armstrong, M., and Sharma, V. (2015). Inferring latent user properties from texts published in social media. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 4296–4297. AAAI Press.

MuNoCC: Multiple Norms Conflict Checker for Multi-Agent Systems

Eduardo Augusto Silvestre¹, Viviane Torres da Silva²

¹Universidade Federal Fluminense (UFF)

²IBM Research (on leave from UFF)

{eduardosilvestre@iftm.edu.br, vivianet@br.ibm.com}

***Abstract.** Norms are being used to regulate the behavior of the autonomous agents. Norms describe the behavior that the agent is obliged, permitted and forbidden to perform in the system. One of the main challenges on developing normative systems is that norms may conflict with each other. Norms are in conflict when the fulfillment of one norm violates the other and vice-versa. This paper presents a tool entitled MuNoCC that is able to detect several kinds of conflicts. Different from other approaches that do only check for conflicts between pairs of norms, this approach is able to check for conflict by considering several norms together. MuNoCC was validated using formal verification.*

1. Introduction

MAS have been gaining great importance in the development of various applications. MAS are autonomous, and heterogeneous societies that can work to achieve common or different goals [Wooldridge 2009].

In order to deal with the autonomy and diversity of interests among different members, the behavior of agents is governed by a set of norms specified to regulate their actions [da Silva 2008]. The norms govern the behavior of agents by defining obligations, prohibitions or permissions. In a multi-agent system (MAS) with many agents with different goals, an important issue to tackle is that norms can conflict with each other. A conflict occurs when norms regulating the same behavior have been activated and are inconsistent [Vasconcelos et al. 2009]. In such cases, the agent is unable to fulfill all activated norms. The conflicts can be direct (all information needed to find out the conflict are explicitly described in the norm) and indirect (needs information about the environment).

Although there are several works that deal with normative conflicts such as [Cholvy and Cuppens 1995], [Elhag et al. 1999], [Vasconcelos et al. 2009] and [da Silva and Zahn 2013], to the best of our knowledge, all those approaches check for conflicts by analyzing the norms in pairs. However, there are conflicts that can only be detected when we consider several norms together.

For instance, let's consider a conflict that can only be detected if norms N1, N2 and N3 are analyzed together. N1 obliges agent A to study math on Monday. N2 forbids agent A to study law on Monday. N3 obliges agent A to study math and law on the same

day. There are no conflicts between the pairs N1-N2, N2-N3 and N1-N3, but when the three norms are analyzed together, we can figure out the conflict.

In this paper, we present a tool, called MuNoCC, that is able to detect direct normative conflicts among multiple norms. Since the analysis of all possible combinations of groups of norms is a NP-complete problem [Shoham and Tennenholtz 1995] [Morales et al. 2014], the *first challenge* of our approach was to deal with this problem. In order to overcome such issue, we have defined several filters to divide the norms in subsets. The *second challenge* of the approach was to be able to check for conflicts among norms having potentially different deontic concepts. This problem was solved by applying deontic transformations [von Wright 1951] normalizing the norms to permissions. Our *third challenge* was to check for the direct conflicts between the normalized norms of the same subset. The solution for this problem was to find out one given situation where all actions being regulated by the permission can be executed. If such situation exists, there is no conflict.

The remainder of this paper is organized as follows. Section 2 presents the background material about norms. Section 3 describes the multiple norm conflict checker. Section 4 presents details about the implementation of MuNoCC tool. Section 5 describes the approach used for formal verification. Section 6 analyzes some of the main related works. Finally, Section 7 states some conclusions and future work.

2. Background

Norms have been vastly used in open MAS to cope with the heterogeneity, autonomy and diversity of interests among the different members. Norms describe the behavior that can be performed, that must be performed, and that cannot be performed.

Our norm definition is inspired on [da Silva Figueiredo et al. 2011]. In [da Silva Figueiredo et al. 2011], the authors analyze the key strategies found in the literature for describing a norm. According to [da Silva Figueiredo et al. 2011], a norm prohibits, permits or obliges an entity to execute an action in a given context during a certain period of time. The difference between our norm definition and the one presented in [da Silva Figueiredo et al. 2011] is the representation of the action being regulated. In [da Silva Figueiredo et al. 2011] the action is represented by a single constant (e.g. pursue, reach, ..). Our representation is more expressive. An action is defined by the name of the action and, optionally, an object where the action will be executed and a list of attributes (with their values).

For our definition of norm, consider the following definitions for sets: the set Nrm is the set of all norms, the set C is the set of all contexts, the set E is the set of all entities, the set A is the set of all actions, the set Cd is the set of all activations and deactivations condition, the set Org is the set of all organizations, the set Env is the set of all environments, the set Ag is the set of all agents and the set R is the set of all roles.

A norm $n \in Nrm$ is a tuple of the form: $(deoC, c, e, a, ac, dc)$, where $deoC$ is a deontic concept from the set $\{O, F, P\}$, respectively, obliged, forbidden and permitted; $c \in C$ is the context where the norm is defined; $e \in E$ is the entity whose action is being regulated; $a \in A$ is the action being regulated; $ac \in Cd$ indicates the condition that activates the norm and $dc \in Cd$ is the condition that deactivates the norm.

Every norm is defined in the scope of a context. The entity, whose action is being regulated, must fulfill the norm when executing in the context where the norm is being defined. In this paper, we consider that a norm can be defined in the context of an organization $org \in Org$ or of an environment $env \in Env$. The set of possible contexts are defined as $C = Org \cup Env$. A norm regulates the action of an agent $ag \in Ag$, an organization (or group of agents) $org \in Org$ or a role $r \in R$. Agents, organizations and roles are entities of the set $E = Ag \cup R \cup Org$.

The activation and deactivation conditions, $ac \in Cd$ and $dc \in Cd$, can state an event that can be a date, the execution of an action, the fulfillment of a norm, etc. In this paper, we will focus on the specification of dates since it is easier to figure out which event has occurred first. Thus, we use simple mathematics symbols such as \leq and \geq to indicate that an event occurs before or after another ($ac \leq dc$).

The four different ways to represent the element action are:

- (i) action;
- (ii) action object;
- (iii) action (attribute1 = {value1}, attribute2 = {value2}, ...);
- (iv) action object (attribute1 = {value1}, attribute2 = {value2} ,...).

The designer of a MAS can set any of the types of norms to a domain, these different ways of defining a norm represents a great flexibility in creating a MAS. In order to exemplify these four ways to describe an action, let's consider the following four prohibition norms:

- (i) Na forbids agent A to study;
- (ii) Nb forbids agent A to study math;
- (iii) Nc forbids agent A to study on Monday
- (iv) Nd forbids agent A to study Math on Monday.

The actions described in the norms are represented as :

- (i) Na: study;
- (ii) Nb: study math;
- (iii) Nc: study (day={Monday})
- (iv) Nd: study math (day={Monday}).

3. Conflict Checker

The main goal of this section is to present the approach that is able to detect direct conflicts among multiple norms. From the set of four types of norms, only norms of type (iv) must be considered when checking for direct conflicts among multiple norms because the structure of the actions used by this type of norms makes possible the identification of objects and their attributes. In order to find conflicts among multiple norms it is necessary to find similarities among the objects of the norms. Norms N1, N2 and N3 (described in Section 1) exemplify this scenario. N3 obliges agent A to study math and law in the same day. Notice that the action is applied over two different objects: math and law. Thus, it is necessary two more norms for analyzing the conflict: a norm that states the possible days for studying math and a norm that states the possible

days for studying law. The relationship among the objects identified in the actions is necessary for the existence of a conflict among multiple norms.

The conflict checker presented in this section will focus on the checking for conflicts among multiple norms, i.e., norms of type (iv) because it is the main contribution of this research. This section is divided in four subsections: strategy, example, algorithms and complexity.

3.1. Applied Strategy

The conflict checker algorithm is divided in three steps. The *first step* is responsible for filtering the norms by including them into sets of similar norms. In order to do so, such step uses three filters. The first filter separates into sets the norms that apply in the same context, the second filter separates into subsets the norms that govern the same entity and the third filter separates into subsets the ones that regulate the same action. After applying all filters, only the norms stored in the same set are the ones that may be in conflict. Norms stored in different sets apply in different contexts, govern different entities and regulate different actions (they do not conflict).

The *second step* of the algorithm is done to solve the problem of analyzing several norms with different deontic concepts at the same time. Our strategy to overcome such problem is to use a single deontic concept to analyze the norms. Note that we do not change the original norms.

Several approaches studied the deontic transformations. Some approaches use the O operator as primitive [McNamara 2014] while others use the P operator [von Wright 1951]. In this work, we use the P operator as primitive and apply the following abbreviations to transform an obligation to a permission (case (1)) and a prohibition to a permission (case (2)): $O_p \leftrightarrow \neg P\neg p$ and $F_p \leftrightarrow \neg Pp$.

From obligation to permission, the authors in [von Wright 1951] proposed the weak axiom $O_p \rightarrow Pp$ that indicates that when p is obliged, p is permitted. Following such axiom and assuming that the designer wants to enable agent A to execute p, we consider that if there is a norm obliging an agent A to execute an action p, such norm can be used as a permission to execute action p. Thus, in this step, all obligations are used as permissions.

From prohibition to permission, this paper uses the Closure Principle which says that what is not explicitly forbidden is permitted [Czelakowski 2015]. Therefore, if there is not a prohibition addressed to an agent to execute an action over an object, the agent is permitted to execute such action over the object. Following this principal, we consider that if there is a norm prohibiting an agent A to execute an action p, such norm states that A is not permitted to execute p. We assume that it is not necessary to create permissions related to everything that is not said in the prohibition. Thus, in this step of the algorithm, all prohibitions are used as negations of permissions.

The checking for conflicts is executed in the *third step* of the algorithm. The algorithm checks if the norms in each set are in conflict. Since all norms are permissions, the analysis made by the conflict checker is very simple; it checks if the norms “intersect”. Two or more norms intersect if there is at least one possible situation where

all the permissions are activated and can be fulfilled, i.e., the actions being permitted can be executed. In such case, the norms are not in conflict.

The conflict checker starts by checking the norms in a set by pairs of norms and then consider all possible set of k -norms until k be equal to the number of norms in the set. At the end, the algorithm has checked for conflicts among all the norms of the set at the same time.

3.2. Running Example

In order to exemplify our approach, let's consider the three norms described in Section 1. We have augmented these norms by including the context where the norms are executed and the periods during while the norms are activated.

N1: Obliges agent A in orgO to study math on Monday in 2016.

N2: Forbids agent A in orgO to study law on Monday in 2016.

N3: Obliges agent A in orgO to study math and law in the same day in 2016.

Norm N3 applies to two objects. It is natural to imagine (without loss of information) that this norm can be divided into two norms: one on math and one on law (strategy to facilitate visualization and checking for conflicts). As the day of study math and law are indifferent, but must be the same day, a variable is used (for convenience, X) and this variable must have the same value for law and math. We get:

N1: (O, orgO, agentA, study math (day = {Monday}, 01/01/2016, 12/31/2016))

N2: (F, orgO, agentA, study law (day = {Monday}), 01/01/2016, 12/31/2016)

N3a: (O, orgO, agentA, study math (day = {X}), 01/01/2016, 12/31/2016)

N3b: (O, orgO, agentA, study law (day = {X}), 01/01/2016, 12/31/2016)

In the first step, the algorithm groups all norms in the same set since they are applied in the same context (orgO), govern action of the same entity (agentA) and regulate the same action (to study). In the second step, the algorithm transforms the norms to permissions. Let's take a look to norm N2 that prohibits agent A to study law on Monday. N2 does not say anything about agent A to study math (in any day), to study on Tuesday or to execute the action of writing papers. In short, the prohibition is just about to study law on Monday. In order to transform a prohibition into a permission, we assume that it is not necessary to create permissions related to everything that is not said in the prohibition (since it is already done by applying the Closure Principle). In addition, any permission that talk about actions and objects that are not the ones refereed in the prohibition are not relevant to the checking of conflicts between the prohibition and any other norm. Therefore, a prohibition like N2 is transformed to a permission that only talks about the agent, action, object and attributes described in the norm. N2 is transformed to a norm that permits agent A to study law NOT on Monday.

Returning to our example, the transformation of norms N1, N3a and N3b into permissions is very simple since they are obligations, as follows:

N1: (P, orgO, agentA, study math (day = {Monday}), 01/01/2016, 12/31/2016)

N3a: (P, orgO, agentA, study law (day = {X}), 01/01/2016, 12/31/2016)

N3b: (P, orgO, agentA, study math (day = {X}), 01/01/2016, 12/31/2016)

The transformation of N2, which is a prohibition, to a permission is done by negating the day of week, that is, the day of week is changed to its complement.

N2: (P, orgO, agentA, study law (day = {NOT Monday}), 01/01/2016, 12/31/2016)

In the third step, the checking for conflicts is executed. In our example of N1, N2 and N3, it is easy to see that any group of two norms is not in conflict. Therefore, it is possible to find out situations where all norms can be fulfilled. The conflict only takes place when we consider the three norms together. In 2016, when the three norms are activated, agentA executing in orgO is permitted to study math on Monday, not study law on Monday and math and law in the same day.

3.3. Algorithms

We now describe the main algorithms used by the conflict checker to detect conflicts among multiple norms. The implementation allows creating norms of types (i), (ii), (iii) and (iv). The program is able to detect direct conflicts using any combination of four types of norms. Due to space limitation, the algorithms described in this paper will focus on the checking for conflicts among multiple norms, i.e., norms of type (iv).

Algorithm 1 receives a set of norms and calls three other algorithms: to classify the norms in sets (Algorithm 2), to convert the norms to permissions (Algorithm 4) and to check for conflicts (Algorithm 5).

Algorithm 2 and Algorithm 3 classify the norms in sets of related norms. Since all norms in a set are similar, i.e., they apply in the same context, govern the same agent and the same action, each new norm is compared with only one of the norms in the set. If the new norm is similar to the norms in the set, the new norm is included in the set. If the new norm is not similar to the norms in the already created sets, a new set is created to store the new norm.

Algorithm 4 is responsible to convert the norms to permissions. In order to transform an obligation to a permission, it is only necessary to rewrite its deontic concept to permission. In order to transform a prohibition to a permission, it is necessary not only to rewrite the deontic concept to permission but also to invert the values of the attributes, as explained in Section A.

Algorithm 5 receives the set of original norms and the sets of the norms converted to permissions. Its main responsibilities are: (i) to check if they can be fulfilled simultaneously, i.e., if they intersect (Algorithm 6), (ii) to create all possible combinations of the converted norms for each set calling function *genAllCombinations* and (iii) to dispatch to check for conflicts (Algorithm 7). To generate all combinations we used an external library of combinatorial analysis.

Algorithm 6 receives a set of norms and checks if there is an intersection among all periods of activation and deactivation. This algorithm works as a new filter because the process only proceeds to the next step if there is a period where all norms are activated at the same time. The further implementation of this filter takes into account their higher cost (necessary to compare periods of activation and deactivation) and the

activation period is analyzed for each combination generated from the norms of the group.

Algorithm 7 receives the sets of norms' combinations and the sets of norms converted to permission. Its main task is to check if they can be fulfilled simultaneously, i.e., if they intersect. This step checks the intersections among the attributes of the norms. If the attributes of the norms do not intersect, there is a conflict among the norms.

3.4. Analysis of Algorithms

The computational cost of the conflict checker is determined by the costs of the calls of the algorithms 2 to 7. The call tree of the algorithms is given by:

Computational Cost *Conflict Checker*

```
(Begin: Algorithm1) conflictChecker (normsSet)
  (Begin: Algorithm2) classifyNormsInSets
    (Algorithm3) normsAreEquivalent
  (End: Algorithm2) classifyNormsInSets
  (Algorithm4) convertNormsToPermissions
  (Begin: Algorithm5) checkForConflicts
    (Algorithm6) normsIntersect
      (Algorithm7) checkMulti
    (End: Algorithm5) checkForConflicts
  (End: Algorithm1)
```

The computation cost is expressed using the big O notation. The letter “n” is the input of norms. Algorithm 3 and Algorithm 4 are linear ($O(n)$). Algorithm 2, Algorithm 6 and Algorithm 7 are polynomials ($O(n^c)$, where c is constant). The Algorithm 5 is the most expensive, since it is exponential ($O(2^k)$). The function *genAllCombinations* makes the algorithm exponential. It was not presented here because it is a well-known combinatorial analysis function. The time and space complexity should be in the order of the number of combinations produced, which is equivalent to the sum of the n -th row of the binomial coefficients in Pascal's triangle or the power set of a set, for example. Thus, the dominant cost of our approach (Algorithm 1) is exponential.

The value of k (complexity of Algorithm 5) comes from the algorithm that groups the norms in sets of similar norms, i.e., norms with the same context, entity and action. The value of k may vary from 1 to n , where n is the number of norms to be checked. Therefore, the best case of the algorithm occurs when $k=1$. In this case, each set of norms stores exactly one norm, i.e., the norms apply in different contexts, govern different entities and actions. In such case, the norms are not in conflict and there is not a need to execute the third step.

The worst case of the algorithm occurs when of $k=n$, i. e., all norms are stored in one set. It can happen if all norms apply in the same context, govern the same entity, and regulate the same action. The cost of the algorithm in the worst case is $O(2^k) = O(2^n)$, where n is the number of norms. The cost of the medium case is $O(2^x)$, where x is the

number of norms in the bigger set. Such evaluation depends on the application domain. Although it is not possible to calculate the medium case, tests showed the power of the filters and we strongly believe that its use can drastically reduce the cost of the conflict checker since it is natural to find in MAS different contexts, entities and actions.

4. MuNoCC (Multiple Norms Conflict Checker Tool)

This section provides details of the tool called MuNoCC. The tool was developed using the Java language and the project is available at <https://goo.gl/QM5Sbe>. There are several types of screens in the application. Some of the most interesting are related to the creation of all the components of the norm. Another interesting feature, which will be addressed in the next paragraphs, is the possibility to transform norms inserted in semi-structured language to norms.

Figure 1 shows the screen where you can import norms and check their conflicts. It shows the norms from Section 3.2 imported and the conflict that the algorithm found. The authors created a compiler to make the process of translating a norm in semi-structured language to Java objects.

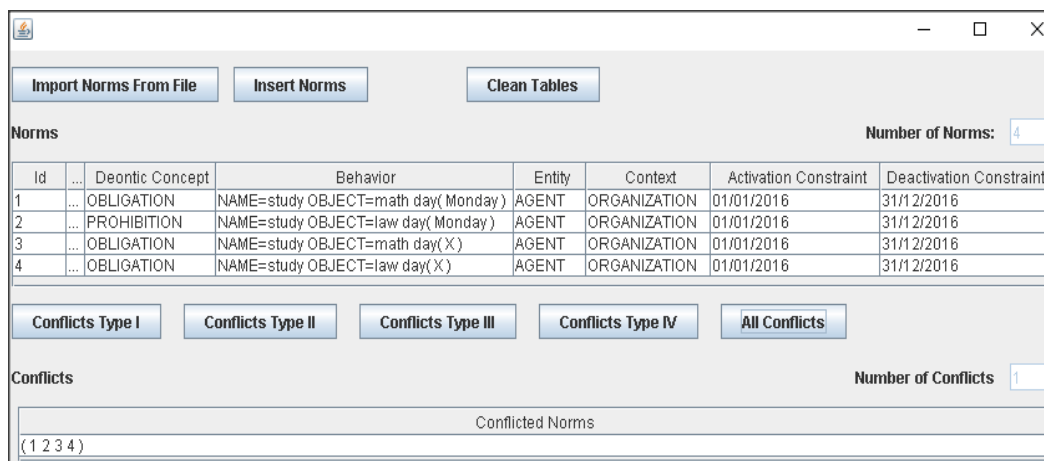


Figure 1. Import norms, parse them and detect conflicts

5. Formal Verification

Although the software testing is the primary means to establish the reliability of software, to carry out exhaustive tests is not feasible even for small software. This section presents the application of formal verification to prove the correctness of the Conflict Checker. This research uses the Design-by-Contract (DBC) [Meyer 1997] to prove the correctness of our approach. The DBC paradigm was applied using the concepts of the JML specification language [Ahrendt et al. 2016]. The implementation of the JML used for checking the correctness of the methods was the KeY tool [Ahrendt et al. 2016]. It can proof the correctness of a program according to its specification. There are two kinds of proofs: partial correctness (do not require the program to terminate, but when terminates is correct) and total correctness (requires that the program terminate).

The classes and methods of the MuNoCC were annotated with the JML features: preconditions, postcondition, loop invariants and class invariants. After annotating the source code with JML, we conducted the proof of the methods by using KeY. To prove

the correctness of algorithms that have loop, one must use the loop invariants in the proof. In JML, a loop specification consists of the following parts: the keyword `loop_invariant` (specifies a loop invariant), the keyword `decreasing` (specifies a value which is always positive and strictly decreased in each loop iteration; it is used to prove termination of the loop) and the keyword `assignable/modifies` (limits the locations that may be changed by the loop).

In order to exemplify our approach, Figure 2 and Figure 3 show parts of the JML added to Java code for a method. The method takes two vectors of any size and returns a resultant vector of the intersection between the two vectors. Figure 2 shows in line 529 that the vectors must not be null. Lines 530 and 531 ensure that all vectors will be executed to its limits (lower limit and upper limit). From lines 532 to 537, the method ensures that the elements belonging to the resulting vector are both setA and setB vectors.

```

528@ /*@ normal_behavior
529 @ requires setA != null && setA.length > 0 && setB != null && setB.length > 0;
530 @ ensures (\forallall int i; 0 <= i && i < setA.length;
531 @ (\forallall int j; 0 <= j && j < setB.length));
532 @ ensures (\forallall int i; 0 <= i && i < \result.length;
533 @ (\exists int k; 0 <= k && k < setA.length;
534 @ \result[i].equals(setA[k])
535 @ &&
536 @ (\exists int j; 0 <= j && j < setB.length;
537 @ \result[i].equals(setB[j])));
538 @*/

```

Figure 2. Example 1 JML set intersection

```

539@ public /*@ pure */ static
540 String[] intersection(String setA[], String setB[]) {
541     int sizeSetA = sizeNotNullD(setA);
542     int sizeSetB = sizeNotNullD(setB);
543     int smaller = (sizeSetA < sizeSetB) ? sizeSetA : sizeSetB;
544     String[] tmp = new String[smaller];
545     int counter = 0;
546     /*@ loop_invariant 0 <= i && i < setA.length;
547     @ modifies i, tmp;
548     @ decreasing setA.length - i;
549     @*/
550     for (int i = 0; i < setA.length; i++) {
551         if (setA[i] != null) {
552             /*@ loop_invariant 0 <= j && j < setB.length;
553             @ modifies j, tmp;
554             @ decreasing setB.length - j;
555             @*/
556             for (int j = 0; j < setB.length; j++) {
557                 if (setB[j] != null) {
558                     if (setA[i].equals(setB[j])) {
559                         tmp[counter++] = new String(setA[i]);
560                     }
561                 }
562             }
563         }
564     }
565     return tmp;
566 }

```

Figure 3. Example 2 JML set intersection

Figure 3 shows the code to perform the intersection between vectors. In line 539, the pure keyword indicates that this method does not modify class variables (only local variables). The two JML blocks, shown in lines 546-549 and 552-555, are very similar. The loop invariants used restrict the allowed values for vectors setA and setB. These kinds of invariants are typical in this scenario. The modifies clause indicates the variables that can be modified in the loop scope.

By considering the execution of Java code annotated with JML, KeY was able to prove the partial correctness of the method by ensuring that the list of actions applied to preconditions implies the postconditions. The KeY tool was also able to prove the total

correctness of the method by ensuring that the variants in lines 548 and 554 decrease in each iteration of the loop and concluding that the algorithm terminates.

6. Related Work

MAS have been gaining great importance in the development of various applications. Several approaches work with normative conflicts in MAS. Some of them deal with the identification of conflicts and others carried out both the identification and resolution of conflicts. However, to the best of our knowledge, all approaches that check for conflicts do only analyze the norms in pairs. There are some papers in the literature that discuss the complexity of considering multiple norms when checking for conflicts, but none of them presents a solution to overcome such problem.

For instance, the works in [Vasconcelos et al. 2009] [Cholvy and Cuppens 1995] [Elhag et al. 1999] [da Silva and Zahn 2013] focus on the identification of direct and indirect conflicts (i.e., conflicts that occur when the elements of norms being analyzed are not the same, but are somehow related). All of these approaches analyze the norms in pairs when checking for conflicts.

Shoham and Tennenholtz [Shoham and Tennenholtz 1995] evaluate the complexity of norm synthesis, i.e., they evaluate the complexity of finding the set of norms that do not conflict. They attested that the problem is NP-complete through a reduction from 3-SAT. In [Vasconcelos et al. 2009] the authors also state that the complexity of inserting or removing a norm in a set of norms is exponential in the worst case. Other approaches that evaluate norm synthesis (such as [Morales et al. 2014]) are based on heuristics and are often intractable in the general case. These studies have corroborated to show that the detection of conflicts among multiple norms is a very complex activity. Therefore, before thinking in a strategy to find the conflicts, it is necessary to think in a strategy to minimize the complexity of the problem, as we have done in our solution to the problem.

7. Conclusions and Future Work

Research in the area of MAS has strongly increased in recent years, in particular, research in normative systems. Despite significant, several research problems have not yet been addressed, there are still many challenges to be considered.

After an extensive literature search, several articles were found on the identification of normative conflicts, which is the topic of this paper. Such works focus on the identification of normative conflicts by considering pairs of norms. However, there are conflicts, as the ones exemplified in the text, which can only be detected when checking for conflicts among multiple norms.

The paper presents an approach able to check for conflicts among multiple norms that uses filters and transformations to reduce the computational cost of the algorithm. The main contributions of the paper are: an extended specification of norms able to regulate different kinds of actions and their objects; the specification of filters to divide the norms in sets and make the checking for conflicts quicker; the deontic transformation to deal with norms having the same deontic concept; a strategy to check for conflicts based on the intersection between norms; a tool to help MAS designer to check for normative conflicts and the proof of correctness of the approach.

A direct consequence of this work is the investigation of how the conflicts among multiple norms should be solved. Can the techniques used to solve conflicts among pairs of norms be used to solve conflicts among multiple norms? An initial approach should investigate the applicability of famous techniques found in literature used to solve conflicts among pairs of norms, such as, *lex posterior*, *lex superior* and *lex specialis*; setting priority among norms and association of a value stating the importance of a norm, should also be investigated for the solving of conflicts.

In this work, we have not considered indirect conflicts [da Silva and Zahn 2013]. The algorithm presented in this paper does only check for direct conflicts, i.e., conflicts among norms that have the same entities, contexts and actions. An important and necessary extension of our work is the identification of indirect conflict among multiple norms.

References

- Wolfgang Ahrendt, Bernhard Beckert, Richard Bubel, Reiner Hahnle, Vladimir Klebanov, and Peter H. Schmitt. 2016. *The KeY Book: Deductive Software Verification in Practice*,
- Laurence Cholvy and Frédéric Cuppens. 1995. Solving Normative Conflicts by Merging Roles. In *Proceedings of the 5th International Conference on Artificial Intelligence and Law*. ICAIL '95. New York, NY, USA: ACM, 201–209. DOI:<http://dx.doi.org/10.1145/222092.222241>
- Janusz Czelakowski. 2015. *Freedom and Enforcement in Action: A Study in Formal Action Theory*, Springer Publishing Company, Incorporated.
- Abdullatif A.O. Elhag, Joost A. Breuker, and Bob W. Brouwer. 1999. On the Formal Analysis of Normative Conflicts. In H.Jaa. van den Herik et al., ed. *JURIX 1999: The Twelfth Annual Conference*. Frontiers in Artificial Intelligence and Applications. Nijmegen: GNI, 35–46.
- Paul McNamara. 2014. Deontic Logic. In Edward N. Zalta, ed. *The Stanford Encyclopedia of Philosophy*.
- Bertrand Meyer. 1997. Design by Contract: Making Object-Oriented Programs that Work. In *{TOOLS} 1997: 25th International Conference on Technology of Object-Oriented Languages and Systems, 24-28 November 1997, Melbourne, Australia*. 360. DOI:<http://dx.doi.org/10.1109/TOOLS.1997.681888>
- Javier Morales, Maite López-Sánchez, Juan A. Rodríguez-Aguilar, Michael Wooldridge, and Wamberto W. Vasconcelos. 2014. Minimality and Simplicity in the On-line Automated Synthesis of Normative Systems. In *AAMAS '14*. Paris, France: IFAAMAS, 109–116.
- Yoav Shoham and Moshe Tennenholtz. 1995. On social laws for artificial agent societies: off-line design. *Artif. Intell.* 73, 1–2 (1995), 231–252. DOI:[http://dx.doi.org/http://dx.doi.org/10.1016/0004-3702\(94\)00007-N](http://dx.doi.org/http://dx.doi.org/10.1016/0004-3702(94)00007-N)
- Viviane Torres da Silva. 2008. From the specification to the implementation of norms: an automatic approach to generate rules from norms to govern the behavior of agents. *Auton. Agent. Multi. Agent. Syst.* 17, 1 (2008), 113–155. DOI:<http://dx.doi.org/10.1007/s10458-008-9039-8>
- Viviane Torres da Silva and Jean Zahn. 2013. Normative Conflicts that Depend on the Domain. In *Coordination, Organizations, Institutions, and Norms in Agent Systems {IX} - {COIN} 2013 International Workshops, COIN@AAMAS, St. Paul, MN, USA, May 6, 2013, COIN@PRIMA, Dunedin, New Zealand, December 3, 2013, Revised Selected Papers*. 311–326. DOI:http://dx.doi.org/10.1007/978-3-319-07314-9_17
- Karen da Silva Figueiredo, Viviane Torres da Silva, and Christiano de Oliveira Braga.

2011. Modeling Norms in Multi-agent Systems with NormML. In *Proceedings of the 6th International Conference on Coordination, Organizations, Institutions, and Norms in Agent Systems*. COIN@AAMAS'10. Berlin, Heidelberg: Springer-Verlag, 39–57.
- Wamberto Weber Vasconcelos, Martin J. Kollingbaum, and Timothy J. Norman. 2009. Normative conflict resolution in multi-agent systems. *Auton. Agent. Multi. Agent. Syst.* 19, 2 (2009), 124–152. DOI:<http://dx.doi.org/10.1007/s10458-008-9070-9>
- Michael Wooldridge. 2009. *An Introduction to MultiAgent Systems* 2nd ed., John Wiley Publishing.
- G.H. von Wright. 1951. Deontic Logic. *Mind* 60, 237 (1951), 1–15.

Otimização de percurso para Plotter Vertical utilizando Algoritmos Genéticos

Caio M. dos S. Escobar¹, Sandro Carvalho Izidoro²

¹Programa de Graduação em Engenharia de Controle e Automação – Universidade Federal de Itajubá (UNIFEI) – Itabira, MG – Brasil

²Programa de Graduação em Engenharia da Computação – Universidade Federal de Itajubá (UNIFEI) – Itabira, MG – Brasil

caioimeschiatti@gmail.com, sandroizidoro@unifei.edu.br

Resumo. *Os plotters verticais são dispositivos capazes de transformar imagens virtuais em reais, idealmente limitadas apenas pelo posicionamento físico do seu sistema de fixação. Suas aplicações são muito restritas atualmente, mas este cenário pode mudar, considerando-se o crescente número de trabalhos e desenvolvimento tecnológico relacionados a este tema. Um dos fatores que restringem suas aplicações deve-se à seu elevado tempo de execução de simples rascunhos. Procurando atenuar este problema, este artigo investiga e avalia a técnica de Algoritmos Genéticos como método de otimização de rotas, encontrando uma rota minimizada única para cada imagem. Este artigo também apresenta uma revisão bibliográfica sobre alguns dos possíveis modelos dos operadores genéticos que realizam o aprimoramento das soluções dentro dos Algoritmos Genéticos, com o passar das gerações.*

Abstract. *Vertical plotters are devices able to convert virtual images into real images, ideally limited only by its fixation system's physical positioning. By the moment they have just a few applications, but this scenario may change very soon considering the increasing number of works and technological development in the field. One of the restricting factors is due to its huge time to perform any simple draft. In an attempt to mitigate this problem, this article explores and evaluates the Genetic Algorithms technique as a route optimizer, finding a minimized unique route for each singular drawing. This article also presents a bibliographic review about some of the possible models for genetic operators that perform the improvement of solutions within Genetic Algorithms, as the generations go by.*

1. Introdução

Um plotter vertical é um dispositivo que possui a premissa de desenhar uma imagem virtual em uma superfície vertical. Segundo Cassimiro e Oliveira (2014), “o Plotter Vertical é uma máquina para desenhar consistindo de uma caneta, suspensa por os entre dois motores, que se move sobre uma superfície vertical.” Se observado mais a fundo, a ferramenta para executar o desenho varia e pode nem sempre ser uma caneta para execução do desenho. Em 2002, Lehni e Franke construíram um plotter vertical que desenha utilizando um spray de tinta. De toda forma, os plotters apresentam-se utilizando diferentes insumos para alcançar o desenho final, e de forma a generalizar o dispositivo que executa o desenho, adota-se aqui o termo “rabiscador”. O rabiscador tipicamente

consiste em um atuador elétrico que permanece suspenso pelo par de cordas e é o que torna possível a diferenciação entre efetuar o desenho ou não (este último é necessário para o reposicionamento do rabiscador).

O funcionamento básico do plotter requer que o rabiscador tenha livre acesso pela superfície por meio de um par de roldanas, que ficam presos nos cantos superiores da área de desenho estipulada. Um par de motores sincronizados controlam os comprimentos de ambas as cordas que suspendem o rabiscador, cada uma de um lado. As cordas que suspendem o rabiscador, são passadas pelas polias fixadas nos cantos superiores, permitindo o posicionamento em toda a área do quadro, ligado aos motores. Pelo menos dois motores elétricos são utilizados, para controlar independentemente as duas extremidades acopladas ao rabiscador. O esquemático do plotter vertical generalizado pode ser observado, conforme exemplo demonstrado na Figura 1.

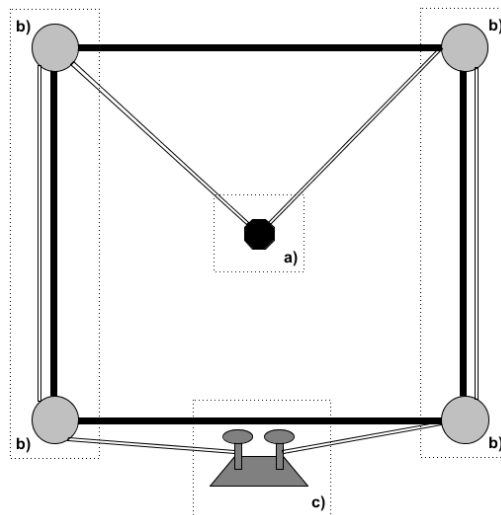


Figura 1. Esquemático de um plotter vertical típico.

Notam-se algumas marcações na Figura 1 representadas por letras, as quais simbolizam as diferentes partes que compõe o plotter vertical. A letra a) representa o “rabiscador” (dispositivo que possua tinta ou similar que possibilite a escrita e/ou desenho) e que fica suspenso por um par de fios. As letras b) representam as polias simples. São quatro no total, e são afixadas nas quatro extremidades da superfície em que se deseja reproduzir o desenho. E finalmente, c) simboliza os motores geralmente motores de corrente contínua (CC) que são responsáveis pela efetiva navegação no quadro. Vale lembrar que esta é uma representação genérica e existem variações das características construtivas do plotter, devido à flexibilidade e simplicidade do conceito. Um dispositivo como tal, permite ao usuário imprimir uma imagem no tamanho desejado, bastando apenas fixar as polias na superfície vertical. Esta construção representa uma evolução das impressoras convencionais, ao passo que este dispositivo pode executar ilustrações de qualquer tamanho em superfícies verticais, desde que não haja obstáculos.

A literatura apresenta autores como Cassimiro e Oliveira (2014), Lehni e Franke (2002), Lehni e Rich (2008), e Noble (2011), que se dedicaram a não apenas executar o desenho, mas executá-lo de forma otimizada. Em função da complexidade do problema, e frente à impraticabilidade da modelagem do tempo de movimentação de todos os plotters existentes, propõe-se neste trabalho, uma otimização de movimentos necessários para a execução do desenho, de forma a economizar consequentemente tempo na

execução final, de uma forma geral para os plotters verticais. Cassimiro e Oliveira ainda sugerem que a decisão do percurso seja feita através de um método eficiente propondo que seria análogo ao Problema do Caixeiro Viajante (PCV).

Guedes (2005) afirma que o PCV “[...] pode ser estabelecido da seguinte forma: dado um conjunto de cidades e o custo da viagem entre cada par delas, procurar encontrar a maneira mais barata de visitar todas as cidades e retornar ao seu ponto de origem, visitando cada cidade exatamente uma vez”. O PCV é um problema de engenharia amplamente conhecido, onde o principal obstáculo é o aumento exponencial do número de soluções. O PCV é um problema que a teoria da computação nomeia NP-Difícil [Morais 1998], ou seja, caracteriza problemas em que “os algoritmos conhecidos para encontrar sua solução exata são intratáveis pelo computador” [Lacerda e Carvalho 1999], o que acaba por inviabilizar algoritmos de busca simples como o algoritmo de força bruta. Neste cenário, métodos de busca heurísticos ganham cada vez mais destaque como, por exemplo, a programação genética através dos Algoritmos Genéticos (AGs).

2. Algoritmos Genéticos

AGs (ou GA, do inglês Genetic Algorithm) formam um conjunto de algoritmos proposto por Holland (1975) e são uma técnica amplamente utilizada na solução de problemas de otimização multiobjetivos. Esta técnica “[...] fornece um mecanismo de busca adaptativa que se baseia no princípio Darwiniano de reprodução e sobrevivência dos mais aptos” [Pacheco 1999]. Este método apresenta resultados positivos quando aplicado em problemas de alta complexidade, problemas inexatos, problemas de programação linear e não-linear utilizando-se da programação evolutiva como ferramenta de otimização, em um espaço de busca consideravelmente amplo. O desenvolvimento dos AGs para problemas de otimização usualmente seguem o diagrama apresentado por Izidoro *et al.* (2015) demonstrado na Figura 2.

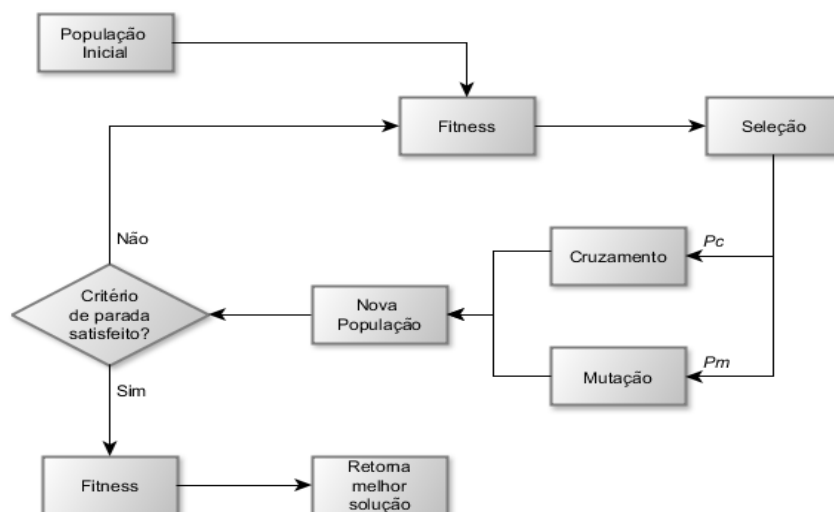


Figura 2. Algoritmo genético padrão. Fonte: Izidoro *et al.* (2015).

Pacheco (1999) sugere que os AGs devem ser aplicados a problemas combinacionais com muitos parâmetros, que necessitam ser recombinados da maneira correta para satisfazer a condição de melhor solução, e Souza *et al.* (2010) apresentam uma otimização multiobjetivo de tomada de decisão em agroindústrias, apresentando resultados atrativos com a técnica. Os AGs apresentam elevada eficácia em encontrar

soluções em recombinações, o que pode ser proveitoso se aplicada em otimização de rotas, provendo um bom planejamento para o rabiscador. Uma das semelhanças entre o PCV com problema de percurso do plotter vertical a ser destacada é que ambos são problemas que geram soluções através de recombinação dos genes. Neste caso, a ordem dos movimentos que o rabiscador deverá executar (equivalente à ordem das cidades no PCV) deve ser recombinada a fim de se encontrar o indivíduo ótimo, que representa a solução de menor movimentação para a plotter (ou para o caixeiro viajante).

Os critérios de parada mais comuns estabelecidos na literatura [Lacerda e Carvalho 1999] são: alcançar um limite pré-estabelecido de gerações, alcançar o valor ótimo (quando conhecido) ou convergência de maior parte dos indivíduos para o mesmo valor. Além do critério de parada, alguns tópicos deste modelo canônico de AGs são aqui divididos em três subseções: Avaliação, Seleção e Reprodução.

2.1. Avaliação

Também chamada de aptidão, ou função *fitness*, é feita através de um cálculo computacional de como ou quanto do(s) objetivo(s) do problema foram alcançados. Esta função depende diretamente do problema, e deve ser analisada e desenvolvida especificamente para cada situação, exercendo papel fundamental no funcionamento dos AGs. Ela é responsável por classificar cada um dos indivíduos da atual população, listando-os por ordem decrescente de aptidão.

Esta classificação é de suma importância, pois exerce grande influência na seleção, aumentando as chances de reprodução das melhores soluções e implicitamente, extinguindo soluções menos eficientes. Tratando-se de um problema como o planejamento de percurso, é intuitivo que a função *fitness* calcule o deslocamento total em uma simulação, e a busca, no sentido de minimização desta função.

2.2. Seleção

Aptidão é a medida do desempenho daquele indivíduo em seu ambiente, também caracterizada pela função objetivo do problema. Esta função é um dos itens mais importantes da otimização, pois qualifica os indivíduos, provendo uma classificação coerente da população. A literatura apresenta alguns métodos mais utilizados e simplistas, e dentre os mais citados, encontram-se os métodos de seleção da roleta e do torneio como observa-se em Lacerda e Carvalho (1999), Pacheco (1999), Souza *et al.* (2010) e Lacerda *et al.* (2002).

O método da roleta fornece aos indivíduos uma chance de reprodução proporcional à sua aptidão, garantindo que os indivíduos mais bem adaptados às condições (restrições do problema), tenham uma chance maior de serem selecionados. A seleção por torneio inicia-se na escolha de um número k de indivíduos, dentro da população atual. Após a seleção de k indivíduos, todos são confrontados entre si, selecionando-se o de maior (se for um problema de maximização) ou menor (caso seja problema de minimização) valor numérico de aptidão dentre os sorteados.

Em geral, o papel do algoritmo de seleção deve ser garantir que a próxima geração de indivíduos seja uma forma de variação dos melhores indivíduos da geração passada, para que seja transmitida a evolução no sentido da otimização desejada, de geração a geração.

2.3. Reprodução

A reprodução é o algoritmo responsável por causar as efetivas variações nos cromossomos, recombina-os em uma nova solução. Durante a reprodução, operadores genéticos como a reprodução (*crossover*) e a mutação são efetuados, recombina os genes e garantindo a diversidade dos indivíduos [Lacerda *et al.* 2002].

Pacheco (1999) cita algumas formas de reprodução, entre elas a troca de toda a população e a troca de população com elitismo, diferenciando-as apenas pela conservação integral de alguns dos melhores indivíduos da geração passada. Lacerda *et al.* (2002) afirmam que “é interessante transferir o melhor cromossomo de uma geração para outra sem alterações. Afinal, porque perder a melhor solução encontrada até então? Esta estratégia é denominada Elitismo.” O Elitismo é uma forma de preservar a solução obtida, incorporando o(s) melhor(es) indivíduo(s) na população da próxima geração.

Algumas formas de recombinação dos cromossomos são sugeridos pela literatura, como o *crossover* de um ponto, *crossover* de n pontos e *crossover* uniforme, que são citados como métodos de recombinação dos dados (cromossomos) de cada indivíduo, a partir de fracionamentos em cabeça e cauda(s) do código genético dos genitores, dando razão ao nome de *crossover* de um ou de n pontos. Em seguida, é sorteada uma nova ordem das seções intercalando-as, gerando um novo indivíduo a partir de um par de soluções. Este método permite a evolução das soluções já encontradas, ao passo que favorece a diversificação.

Além do operador de recombinação, há ainda o operador de mutação que é incumbido de fazer inversões em genes aleatórios dos cromossomos dos filhos. Lacerda e Carvalho (1999) citam que a “mutação melhora a diversidade dos cromossomos na população, no entanto por outro lado, destrói informação contida no cromossomo, logo, deve ser utilizada uma taxa de mutação pequena (normalmente entre 0,1% a 5%), mas suficiente para assegurar a diversidade”.

3. Metodologia

Entre o pré-processamento dos dados que compõem as imagens e a evolução das soluções, foram necessárias algumas considerações, desde escolhas básicas de reconstrução do desenho, até a modelagem dos indivíduos e operadores genéticos para a organização da estrutura dos AGs. Os métodos de representação do problema desenvolvidos neste artigo são descritos nas subseções seguintes.

3.1. Considerações Gerais

Para a reprodução do desenho na superfície, em primeiro lugar, o mesmo deve ser redimensionado através de técnicas numéricas que permitem interpolar os dados da imagem. O par de motores responsáveis pelo posicionamento na superfície devem atuar coordenadamente para que o rabiscador seja colocado diretamente em cima de cada um dos pontos proporcionais aos que devem ser pintados do desenho original, um por vez.

A princípio, há duas maneiras de se reconstruir o desenho: por meio de pontos, ou de linhas. O algoritmo expandido de funcionamento e suas diferenças podem ser analisados por meio da Tabela 1.

Tabela 1. Métodos básicos para a reprodução de imagens.

Método 1 – Pontos	Método 2 – Vetores
Posiciona em #1.	Posiciona em vetor #1.
Aciona rabiscador.	Aciona rabiscador.
Recua rabiscador.	Efetua o vetor #1.
Posiciona em #2.	Recua rabiscador.
Aciona rabiscador.	Posiciona em vetor #2.
Recua rabiscador.	Aciona rabiscador.
....	...
Aciona rabiscador.	Efetua o vetor #n.
Recua rabiscador.	Recua rabiscador.

A vantagem da implementação do Método 2 descrito pela Tabela 1 se manifesta em desenhos que contenham muitos pixels para serem representados e/ou estes se apresentam congruentes. Se o desenho possuir pixels agregados como na Figura 3.a, o método de representação por linhas apresentará maior eficácia por evitar movimentos desnecessários. Por outro lado, se o desenho contiver pixels isolados, como demonstrado na Figura 3, o Método 2 resultará no mesmo número de movimentos apresentado no Método 1. Neste caso, o pixel será representado por uma reta de comprimento igual a um. Isto justifica a aplicação do método de representação por linhas no presente trabalho.

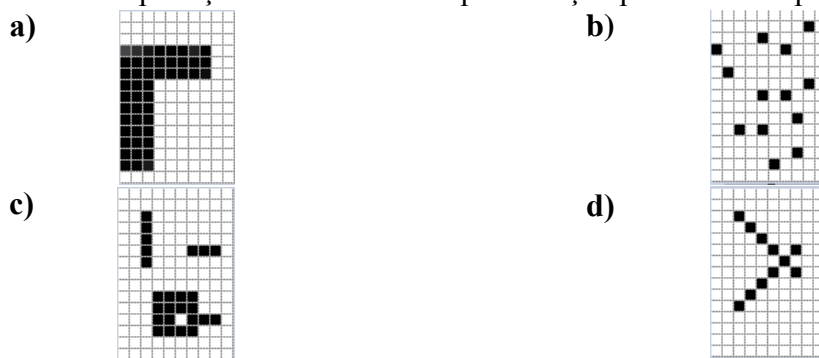


Figura 3. a) Exemplo de pixels congruentes. b) Exemplo de pixels isolados c) Exemplo de desenho com pixels considerados congruentes. d) Exemplo de desenho com pixels considerados isolados.

Para que o algoritmo não assumisse uma complexidade muito elevada, adotou-se como padrão que os pixels em diagonal não são considerados congruentes, apenas horizontal ou vertical, como demonstra-se nas Figura 3.c e Figura 3.d. A economia de movimentos tem impacto na redução de tempo para execução do desenho completo, viabilizando a reconstrução do desenho através do Método 2, descrito na Tabela 1. Isso exige que o desenho seja pré-processado para que a informação contida no desenho seja convertida em linhas, para a decorrente reconstrução do desenho. Este procedimento executa a transformação simplificada dos dados do desenho em um processo chamado vetorização da imagem.

Segundo L. Wenyin e D. Dori (1999, p. 10–21, *apud*. Santos, 2009, p. 8), “Quando uma imagem é representada digitalmente (escaneada, fotografada, etc) normalmente assume um formato matricial ou raster. Outra representação possível é a vetorial, que consiste de um conjunto de elementos independentes ou “objetos vetoriais” que possuem formato, posição e tamanho próprios”. Um benefício decorrente da vetorização é que “representações vetoriais mantêm a qualidade independentemente do seu tamanho. Isso acontece porque quando são aumentadas, os pontos que definem o objeto vetorial são recalculados pela função matemática que o define” [Santos 2009]. Reorganizar os vetores

que compõem o desenho de forma ótima, é interessante para que percorra-se o menor caminho possível, poupando-se energia e tempo. Para tal, os vetores são classificados por tamanho, eliminando-se possíveis retas menores que passariam por um ponto já visitado anteriormente. Cada ponto será representado unicamente pela reta (horizontal ou vertical) de maior comprimento, o que garante que cada pixel seja visitado apenas uma vez, assim como cada um dos vetores que compõem o desenho.

3.2. Estrutura e parâmetros dos AGs

Os AGs foram desenvolvidos inteiramente na plataforma Matlab, devido ao grande número de ferramentas que o software fornece para auxiliar no desenvolvimento de algoritmos, e na posterior simulação numérica dos resultados. Neste ambiente, cada indivíduo foi definido como uma matriz de coordenadas, onde são armazenados os valores de cada um dos vetores obtidos da vetorização da imagem. Observa-se na Figura 4, uma representação canônica de um indivíduo.

	$X_{inicial}$	$Y_{inicial}$	X_{final}	Y_{final}
gene #1	$G_{1,1}$	$G_{1,2}$	$G_{1,3}$	$G_{1,4}$
gene #2	$G_{2,1}$	$G_{2,2}$	$G_{2,3}$	$G_{2,4}$
gene #3	$G_{3,1}$	$G_{3,2}$	$G_{3,3}$	$G_{3,4}$
...
gene #(n-1)	$G_{(n-1),1}$	$G_{(n-1),2}$	$G_{(n-1),3}$	$G_{(n-1),4}$
gene #n	$G_{n,1}$	$G_{n,2}$	$G_{n,3}$	$G_{n,4}$

Figura 4. Representação de um indivíduo.

O modelo de indivíduo presente na Figura 4, é nada mais que uma lista de vetores. Cada linha (gene #n) representa um vetor onde o plotter deve efetuar um traço, de $(X_{inicial}, Y_{inicial})$ até (X_{final}, Y_{final}) . Ao iniciar os AGs, uma população com p indivíduos será gerada, com todos os vetores ordenados aleatoriamente. O percurso da plotter é calculado como sendo a sequência de traços descrito pelo indivíduo, e seu *fitness* consequentemente é a somatória total deste percurso. Como trata-se de um problema de minimização, sempre será escolhido o indivíduo com o menor valor de aptidão. O processo segue com a avaliação da população por meio da função *fitness* descrita na equação (1).

$$t_t = \sum_{k=1}^{n-1} \sqrt{(X_i(k+1) - X_f(k))^2 + (Y_i(k+1) - Y_f(k))^2} \quad (1)$$

Onde:

X_i : $X_{inicial}$

Y_i : $Y_{inicial}$

t_t : percurso total (em px)

X_f : X_{final}

Y_f : Y_{final}

n : número de genes

A função *fitness* apresentada em (1) efetua o cálculo da somatória dos módulos de cada um dos vetores de reposicionamento da plotter vertical. Em seguida, inicia-se a elaboração da nova geração, e o elitismo age garantindo que os indivíduos mais bem posicionados do ranking sejam passados diretamente para a próxima geração, sem nenhuma alteração. Após isso, inicia-se o processo de seleção (por torneio) dos indivíduos que preencherão os lugares vagos na nova geração. Selecionado o par de indivíduos genitores, eles são então copiados para a próxima geração e sofrem a reestruturação estabelecida pelo *crossover* de n pontos, conforme exemplo na Figura 5.

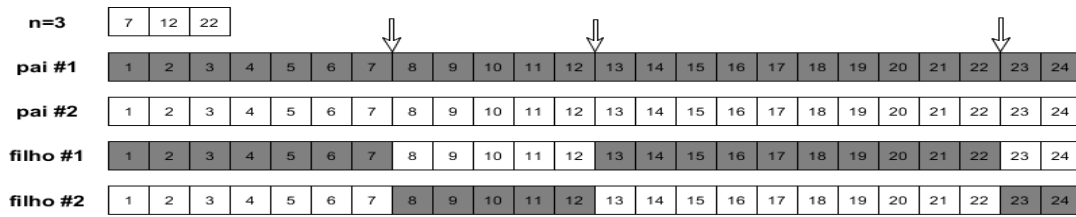


Figura 5. Exemplo de *crossover* de 3 pontos.

O *crossover* de n pontos exemplificado pela Figura 5, mostra um caso particular em que $n=3$, mas que pode ser estendido a qualquer número inteiro dentro do intervalo de genes que constituem um indivíduo. Na Figura 5 observa-se o sorteio de 3 valores inteiros que foram escolhidos dentro do intervalo $[1:24]$ (no exemplo 7, 12 e 22). Os genes do par de pais são permutados entre si, resultando em um novo par de filhos, com características de ambos os pais mescladas. Kumar *et al.* (2012) apresenta este tipo de *crossover* de n pontos sob o nome de PMX (*Partially Matched Crossover*), concluindo que este método é muito efetivo no PCV. Este método baseia-se na permutação dos indivíduos, não necessitando portanto, correções consequentes. Este método de reprodução foi implementado iterando-se até que sejam gerados indivíduos suficientes para preencher toda a população da nova geração.

O operador genético de mutação é aplicado à população, dispensando apenas os indivíduos que foram transferidos através do Elitismo. A mutação implementada consiste de uma simples inversão entre as coordenadas de início e fim de um vetor, em outras palavras, a inversão de um gene. conforme apresentado na Figura 6.

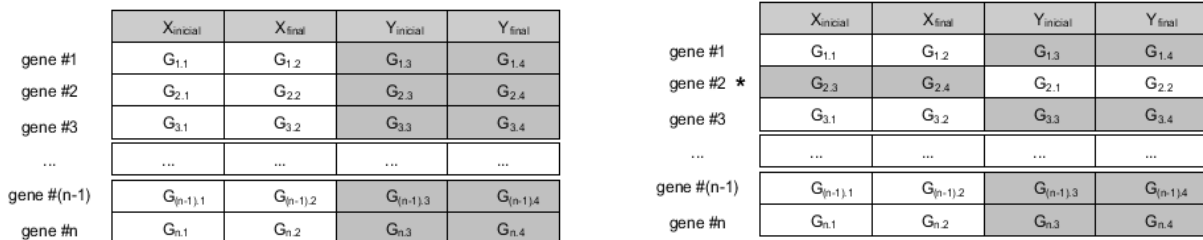


Figura 6. Exemplo de mutação no gene #2.

A Figura 6 é um exemplo da mutação implementada, em que são representados dois momentos de um mesmo indivíduo (antes e depois da mutação), que teve o gene #2 sorteado pelo operador genético. À esquerda, encontra-se o indivíduo inicialmente com todos os genes G ordenados, e ao lado direito verifica-se o mesmo indivíduo transformado por uma única mutação. Esta mutação permite um novo parâmetro na busca da otimização do problema, no sentido de que a solução ótima pode ser encontrada em indivíduos com diferentes combinações de orientação de início e fim dos vetores, além da simples recombinação dos genes.

4. Resultados Experimentais

Com o propósito de se verificar se a disputa entre indivíduos realmente conduz à evolução da solução observada, independente de “sorte ou azar”, foram executadas trinta iterações com cada *dataset* (conjunto de dados). Para tal, foi utilizado um *dataset* proposto por Reinalt (1991) chamado ATT48.

Complementarmente, um *dataset* de figuras foi desenvolvido especialmente para verificar a eficácia do método proposto. Ele foi idealizado de uma forma que seja simples de se encontrar o valor ótimo, mas para os AGs representa uma procura extensa pelo espaço de busca. É uma forma simples e eficaz de realizar a verificação dos resultados encontrados pelos AGs, uma vez que o valor de custo ótimo é conhecido e pode ser usado como valor de referência.

Para a obtenção dos resultados, foram definidos os parâmetros dos AGs de forma empírica. Alguns dos parâmetros foram alterados entre o PCV e o *dataset* próprio, pois foram observados melhores resultados com as devidas alterações. Todos os resultados foram gerados utilizando-se os parâmetros exibidos na Tabela 2.

Tabela 2. Parâmetros utilizados nos AG's.

Parâmetros	ATT48	Dataset figuras
População (indivíduos):	300	300
Selecionados no torneio (indivíduos):	10	10
Elitismo (indivíduos):	2	4
Crossover de n pontos (genes):	3	2
Mutação (% dos genes):	2	5
Máximo de gerações:	300	150

4.1. ATT48

Este *dataset* possui as coordenadas de 48 cidades capitais dos Estados Unidos, e trata-se de um dos mais famosos *datasets* reais do PCV. Muitos autores trabalham com este *dataset* até os dias atuais, principalmente para demonstrar a eficácia de algoritmos de busca como em [Ma e Simon 2017] e [Jain e Prasad 2017]. Morais (2010) apresenta o valor de ótimo do *dataset*, demonstrado na Tabela 3, e ainda, afirma que o tempo estimado para solução utilizando o algoritmo força bruta, seria da ordem de 10^{53} segundos, confirmando sua inviabilidade de implementação e portanto, a necessidade de algoritmos mais sofisticados para a otimização de rotas.

Os resultados obtidos durante a implementação do presente modelo de AG's são exibidos na Tabela 3, podendo ser comparados em valores absolutos com outros trabalhos que também utilizam o ATT48 como referência como em [Morais 2010] e [Kok *et al.* 2012]. Isto assegura que os AGs desenvolvidos para a otimização da *plotter*, apesar de partir de soluções aleatórias, minimizam significativamente o percurso percorrido para a reprodução do desenho em sua escala final.

Tabela 3. Resultados de 30 iterações com o ATT48.

Solução ótima	Melhores indivíduos			Desvio do ótimo
	Média	σ^2	σ	
33523.70	38367	9422159.14	3069.55	14.45%

4.2. Dataset de figuras

Este *dataset* é composto de figuras binárias quadradas de tamanho total igual a cem pixels. Elas permitem verificar se os AGs têm a capacidade de inverter as retas necessárias, ao mesmo tempo que permuta a ordem do desenho. Para atingir o valor ótimo, é necessária movimentação de um único pixel para cada reta extra representada, percorrendo um “zig-zag”. Ou seja, para três vetores paralelos e congruentes, a solução ótima tem um custo (reposicionamento) de 2 pixels. Para quatro vetores paralelos e congruentes, o custo

mínimo é 3, e assim se procede até cem vetores. A representação do *dataset* pode ser observada na Figura 7.

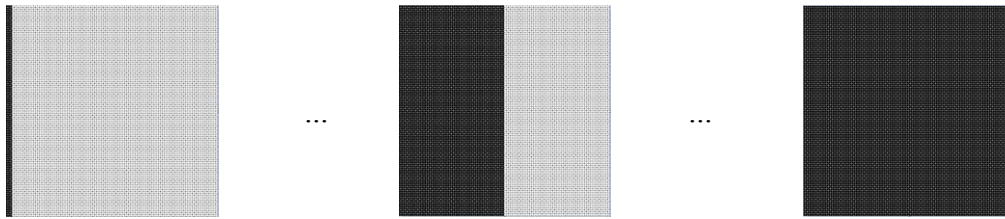


Figura 7. Representação do *dataset* de figuras a) Quatro vetores (esquerda); b) Cinquenta vetores (meio); e c) Cem vetores (direita).

A Figura 7 (a, b e c) apresenta o *dataset* de imagens, que possuem quatro, cinquenta e cem vetores, respectivamente. Outras interpolações destas figuras também foram testadas, e alguns dos resultados obtidos são apresentados na Tabela 4, ressaltando-se que o valor da solução ótima é sempre o número de vetores da figura menos um.

A Tabela 4 apresenta os resultados referentes ao *dataset* de imagens, com o custo de deslocamento do percurso ótimo destacado em negrito. O algoritmo força bruta, também conhecido como busca por exaustão, apresenta sempre a solução ótima, uma vez que explora-se todos os percursos possíveis, mas por outro lado tem um tempo de processamento exponencial. Com o aumento do número de vetores, este algoritmo mostra-se infactível por depender de um tempo muito grande de processamento. As colunas sob “Melhores indivíduos” retratam a média e desvio padrão das melhores soluções obtidas ao longo de trinta iterações dos algoritmos genéticos. Em seguida, “Desvio do ótimo” apresenta o desvio da solução ótima, que apresenta redução mas não garante solução do problema. Em contrapartida, o tempo de processamento dos AG’s apresenta um aumento bem menos significativo, se comparado com o algoritmo força bruta. Na última coluna intitulada “Evolução”, compara-se a média dos melhores indivíduos da última geração iterada com a média dos melhores indivíduos da primeira geração (solução aleatória não otimizada). Esta comparação é feita pois a partir do momento que o dispositivo terá a finalidade de desenhar uma imagem aleatória, um algoritmo de execução estático pode apresentar um desempenho também aleatório.

Tabela 4. Resultados de 30 iterações com o *dataset* de figuras*.

Número de vetores	Solução ótima	Força Bruta		Algoritmos Genéticos			Evolução	
		Desvio do ótimo	Tempo de processamento	Melhores indivíduos	Desvio do ótimo	Tempo de processamento		
				Média	σ			
4	3	0%	0.02 s	3	0	0%	21.75 s	3.36%
5	4	0%	0.04 s	4.13	0.19	3.45%	21.83 s	3.47%
6	5	0%	0.07 s	5.24	0.49	4.87%	22.42 s	3.51%
7	6	0%	0.16 s	7.11	0.96	18.66%	22.60 s	3.97%
8	7	0%	0.36 s	8.60	1.29	22.87%	23.75 s	4.10%
9	8	0%	0.79 s	11.17	2.11	39.63%	24.69 s	4.14%
10	9	0%	1.71 s	12.69	2.17	41.10%	25.80 s	4.23%
11	10	0%	3.71 s	13.13	2.21	31.03%	26.54 s	4.46%
50	49	0%	4.84 10 ¹³ s **	371	66.18	657%	56.51 s	12.70%
100	99	0%	3.15 10 ³⁰ s **	2252	215.68	2174%	88.64 s	22.57%

* Executado em um computador com processador AMD A8-6410 2.0 GHz, 8 GB RAM, Windows 8 64-bits.

** Tempo estimado utilizando-se a “Curve Fitting Tool” do Matlab.

Analisando-se os resultados expostos na Tabela 4, nota-se que apesar do algoritmo apresentar um grande desvio do ótimo, este representa um método de otimização global garantindo melhoria para qualquer imagem de entrada. Normalmente, constata-se que são implementados algoritmos estáticos, como pode ser observado em Noble (2011) e Landoni (2016). Para uma análise comparativa do desempenho do percurso final obtido pelos AG's, foi calculado o desvio do ótimo para o *dataset* exibido na Figura 7, também para um algoritmo estático como o das impressoras tradicionais. O desvio obtido é de 9700.51% e permanece constante para todas as figuras do *dataset* testado, pois todos os vetores possuem orientações iguais e para cada novo vetor, é adicionado um novo movimento de reposicionamento de 98.01 pixels, valor este referente ao incremento de uma linha e retorno do rabiscador à primeira coluna.

5. Conclusão

Atualmente, os AGs são uma das técnicas mais aplicadas em problemas de otimização, considerando que esta também pode ser implementada em conjunto com métodos mais complexos, como redes neurais, autômatos celulares e aprendizado de máquina. Os resultados aqui expostos representam apenas uma ínfima parcela dentre as muitas possibilidades de operadores genéticos e diferentes modelagens que os AGs podem assumir. Além disso, a compreensão do problema, respectiva modelagem dos indivíduos e função *fitness* exercem papéis determinantes no desempenho dos AGs.

Apesar dos resultados demonstrados ainda possuírem um valor considerável de desvio da solução ótima, pode-se afirmar que a otimização por AGs apresentam melhoria relevante quando comparado com algoritmos de execução estáticos. Ao passo que a figura de entrada é aleatória (qualquer figura), é necessária a adoção de uma estratégia abrangente para otimização global da movimentação das plotters verticais.

O método aqui expresso busca gerar uma solução customizada para cada imagem, de forma a reduzir os custos referentes ao consumo de energia elétrica, desgaste mecânico e tempo. Suplementarmente, este trabalho visa uma vulgarização dos plotters verticais, que são dispositivos que tem grande potencial para aplicações cotidianas. Eles executam as funções que uma impressora de mesa pode oferecer, com o diferencial de que permite executar um desenho tão grande quanto se faça necessário. Este sistema tem potencial de substituir tarefas que são atualmente executadas somente por humanos, como por exemplo, desenhos em grandes painéis e murais, ou até mesmo tarefas perigosas, como pinturas prediais e residenciais, onde trabalhadores estão constantemente expostos à riscos de acidentes.

Referências Bibliográficas

- Cassimiro, A. G. e Oliveira, L. B. (2014) "Plotter Vertical", [http://paginapessoal.utfpr.edu.br/msergio/portuguese/ensino-de-fisica/oficina-de-integracao-ii/Luisa-Andre-Etapa Final do Projeto de Oficina de Integracao 2.pdf/view](http://paginapessoal.utfpr.edu.br/msergio/portuguese/ensino-de-fisica/oficina-de-integracao-ii/Luisa-Andre-Etapa%20Final%20do%20Projeto%20de%20Oficina%20de%20Integracao%202.pdf/view).
- Guedes, A. C. B., Leite, J. N. F. e Aloise, D. J. (2005) "Um algoritmo genético com infecção viral para o problema do caixeiro viajante" In:Revista PubliCa, v. 1, n. 1.
- Holland, J. H. (1975) "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence" Cambridge: University Of Michigan Press, 1975. 183 p.

- Izidoro, S. C., Melo-Minardi, R. C. e Pappa, G. L. (2015) "GASS: identifying enzyme active sites with genetic algorithms" In: *Bioinformatics* 2015; 31 (6): 864-870.
- Jain, V. e Prasad, J. S. (2017) "Solving Travelling Salesman Problem Using Greedy Genetic Algorithm GGA" In: *International Journal Of Engineering And Technology (IJET)*. Palwal, p. 1148-1154.
- Kok, J. et al. (2012) "Computational experiments involving population size for FPGA-based implementation of a GA for the TSP" In: *4th international conference on computational methods (ICCM2012)*, November.
- Kumar, N. et al. "A Comparative Analysis of PMX, CX and OX Crossover operators for solving Travelling Salesman Problem" In: *International Journal Of Latest Research In Science And Technology*. Kurukshetra - India, p. 98-101. jul. de 2012.
- Lacerda, E. G. M. e Carvalho, A. C. P. L. F. (1999) "Otimização de redes neurais RBF usando algoritmos genéticos e sua aplicação na área financeira" [s.n.], São Carlos.
- Lacerda, E. G. M., Carvalho, A. C. P. L. F. e Ludermir, T. B. (2002) "Um tutorial sobre algoritmos genéticos" In: *Revista de Informática Teórica e Aplicada*, v. 9 n. 3, p. 109-139, Janeiro.
- Landoni, B. (2016) "How to make an OpenSource Vertical Plotter" Publicado em 23 Jun. <https://www.open-electronics.org/how-to-make-an-opensource-vertical-plotter/>.
- Lehni, J. and Franke, U. (2002) "Hektor" http://www.multimedialab.be/doc/art/doc_hektor.pdf, September.
- Lehni, J. and Rich, A. (2008) "Viktor" In: *A recent history of writing and drawing* [9 jul. - 31 ago.]. Institute Of Contemporary Arts, London. p. 1 - 9. https://www.ica.art/sites/default/files/downloads/ica-recent-history-gallery-guidewVB1WP_0.pdf, August.
- Ma, H. and Simon, D. (2017) "BBO for Combinatorial Optimization" In: *Evolutionary Computation with Biogeography-based Optimization*. Cap. 8, p. 162.
- Morais, J. L. M. (2010) "Problema do Caixeiro Viajante Aplicado ao Roteamento de Veículos numa Malha Viária". Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo, São José dos Campos. <http://www.ft.unicamp.br/docentes/meira/publicacoes/2010jose.pdf>, Dezembro.
- Morais, M. J. C. (1998) "PCV: Um velho problema revisitado estatisticamente" In: *Actas da V Conferência do CEMAPRE*, p. 439-458.
- Noble, S. (2011) "What's a polargraph?: Polargraph? Polarshield? PolargraphSD? Huh?" <http://www.polargraph.co.uk/whats-a-polargraph/#comment-14012>.
- Pacheco, M. A. C. (1999) "Algoritmos genéticos: Princípios e aplicações" <http://www2.ica.ele.puc-rio.br/Downloads\38/CE-Apostila-Comp-Evol.pdf>, Julho.
- Reinelt, G. (1991) "TSPLIB - A Traveling Salesman Problem Library" In: *Journal on Computing*. Online, p. 376-384. November.
- Santos, J. A. (2009) "Reconhecimento semi-automático e vetorização de regiões em imagens de sensoriamento remoto". Universidade Estadual de Campinas, SP. <http://repositorio.unicamp.br/jspui/handle/REPOSIP/276150>, Março.
- Souza, C. C. *et al.* (2010) "Uso de algoritmos genéticos como ferramenta auxiliar no processo decisório em atividades de gestão agroindustrial" In: *Informe GEPEC*, Toledo, v. 14, n. 1, p. 113-126, jan./jun.

Modelagem de Funções de Transferência de Plantas Industriais em Malha Aberta e Fechada utilizando Algoritmos Genéticos

Florisvaldo C. Bomfim Jr^{1,2}, Igor S. Peretta², Keiji Yamanaka²

¹Universidade de Uberaba (UNIUBE)
Uberaba, MG – Brazil

²Faculdade de Engenharia Elétrica
Universidade Federal de Uberlândia (UFU)
Uberlândia, MG – Brazil

florisvaldo.bomfim@gmail.com, {iperetta, keiji}@ufu.br

Abstract. *In control systems, the knowledge of the transfer function of an industrial plant affects the quality of the desired control. When such function is unknown, the engineer must adjust its controllers empirically. This work proposes the use of genetic algorithms to estimate the transfer function of a given industrial plant. The achieved results are compared to results of conventional methods for a fictitious open-loop plant. The proposed method, implemented in MATLAB®, is also able to estimate the closed-loop transfer function even with a poor control.*

Keywords: *Genetic Algorithm, Control, Automation, Transfer Function, Industrial Plant.*

Resumo. *Em sistemas de controle, conhecer a função de transferência de uma planta industrial afeta a qualidade de controle desejado. Quando desconhecida, o engenheiro deve ajustar empiricamente seus controladores. Este trabalho propõe o uso de algoritmos genéticos para estimar a função de transferência de determinada planta industrial. Os resultados obtidos foram confrontados com métodos convencionais para uma planta fictícia em malha aberta. O método proposto, implementado em MATLAB®, consegue ainda estimar a função de transferência em malha fechada mesmo com um controle não ideal.*

Palavras-chave: *Algoritmo Genético, Controle, Automação, Função de Transferência, Planta Industrial.*

Introdução

Em sistemas de controle industrial, conhecer a função de transferência de uma planta pode ser a diferença na qualidade de seu controle. Em diversos casos em produção, tal função de transferência não é conhecida, restando ao engenheiro a tarefa do ajuste empírico de seus controladores. Este tipo de ajuste não otimizado leva a prejuízos na indústria.

A função de transferência é a relação de pelo menos duas variáveis de processo, sendo uma de entrada e outra de saída. Por exemplo, uma função de transferência de uma torneira elétrica pode ter como variável de entrada a vazão da água e como variável de saída a temperatura desejada para a água no final do processo.

Atualmente, há várias formas de se estimar a função de transferência de um sistema, sendo que na maioria dos métodos existentes, é buscada uma aproximação para uma função de transferência de primeira ordem com atraso, que é representada na Equação (1):

$$F(s) = \frac{K \exp(-\theta s)}{\tau s - 1} \quad (1)$$

onde K representa o ganho do sistema, θ o seu atraso e τ é a constante de tempo. Desta forma, nem sempre a função encontrada se aproxima da função real que pode ser de ordens superiores. Essa diferença na função de transferência pode acarretar em um cálculo equivocado dos parâmetros do controlador desejado.

O presente trabalho tem como objetivo apresentar uma forma não convencional de se estimar a função de transferência, a heurística dos algoritmos genéticos (AG). O algoritmo proposto é capaz de estimar funções para sistemas tanto em malha aberta quanto em malha fechada. Sistemas em malha aberta são caracterizados pelo fato de sua variável controlada não ser realimentada. Já os sistemas de malha fechada são realimentados em sua entrada com informações da saída, podendo se estabilizar em algum ponto de interesse se bem controlados. O método apresentado consegue, ao estimar uma função de transferência mais próxima da planta real, possibilitar aos engenheiros responsáveis um melhor projeto de controle.

O foco deste trabalho é a modelagem de uma planta industrial através da estimativa de sua função de transferência: (i) Quando em malha aberta, o resultado deste trabalho deve ser usado para o cálculo mais apropriado dos parâmetros do controlador PID (Proporcional, Integral e Derivativo); (ii) Quando em malha fechada, o resultado deste trabalho pode ser usado para verificação e possível correção dos parâmetros do controlador PID em produção, visando o aperfeiçoamento do desempenho da planta.

Trabalhos relacionados

O desejo de automatizar o modelamento de sistemas não é recente. Funções de transferência podem descrever sistemas e permitir seu controle. Trabalhos como o visto em [He et al. 1996] utiliza técnicas de busca estocásticas para gerar funções de transferência. Em [Loyka 1999], existe a aplicação de algoritmos genéticos para modelar sistemas não lineares.

Trabalhos como os de [Nedelea et al. 2006] e de [Cabral and de Melo 2011] também se utilizam de algoritmos genéticos relacionando-os com funções de transferências, mas o objetivo de ambos é o modelamento de filtros, um sistema específico que difere deste presente trabalho. Ainda sobre aplicações, em [McCall 2005] tem-se funções de transferência através de modelamento por otimização matemática no campo da imunologia. O trabalho de [Tavakolpour et al. 2010] foca em modelamento de sistemas com função de transferência, mas não trabalha com atrasos ou aplicações industriais.

Em [Biradar et al. 2016], seus autores se preocupam com sistemas lineares invariantes no tempo, o tipo de sistemas para o qual este presente trabalho é focado, mas que difere em aplicações.

O método aqui apresentado traz contribuições que diferem dos trabalhos relacionados encontrados, uma vez que considera que o sistema a ser modelado pode ditar a

ordem da função de transferência, além de trazer comparações com os métodos consagrados de modelamento de sistemas. As aplicações de interesse são plantas industriais em produção que possam ser amostradas em malha aberta ou em malha fechada.

Fundamentos para a investigação

Nesta seção são discutidos conceitos importantes para o entendimento da proposta apresentada neste trabalho.

Função de Transferência

A função de transferência, descrita como a relação da variável de entrada com a variável de saída, para um sistema de equações diferenciais lineares é definida como a relação da Transformada de Laplace da saída com a Transformada de Laplace da entrada. Considere a Equação (2):

$$\sum_{i=0}^n a_i \frac{d^i}{dt^i} y = \sum_{j=0}^m b_j \frac{d^j}{dt^j} x \quad (2)$$

onde $y(t)$ é a saída do sistema e $x(t)$ sua entrada. Note que $n \leq m$.

Aplicando as respectivas transformadas de Laplace \mathcal{L} na equação diferencial em função do tempo, temos a Equação (3) no domínio s da frequência:

$$\begin{aligned} G(s) &= \frac{\mathcal{L}[y(t)]}{\mathcal{L}[x(t)]} = \frac{Y(s)}{X(s)} \\ &= \frac{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0} \cdot \exp(-\theta s) \end{aligned} \quad (3)$$

Note que uma função de transferência de um sistema linear é o conjunto de parâmetros invariantes no tempo.

Tipos de malhas

Em sistemas de controle, pode-se colocar uma malha de controle em duas condições possíveis: em malha aberta (ver Figura 1), e em malha fechada (ver Figura 2). Em sistemas de malha aberta não há atuação do compensador (controlador) em relação à variável de saída. Dessa forma, pode-se dizer que não existe uma retroalimentação. Já em sistemas de malha fechada tem-se a comparação da variável controlada $y(t)$ com a variável de referência $r(t)$, gerando assim um valor de erro $e(t)$, conforme apresentado na Equação (4), que será usado na compensação da saída.

$$e(t) = r(t) - y(t) \quad (4)$$

Modelos matemáticos

Os sistemas físicos reais apresentam uma alta complexidade o que dificulta em muito o seu estudo. Com isso, a dificuldade da modelagem está diretamente relacionada à natureza e à precisão do estudo que se deseja fazer referente ao comportamento do sistema.

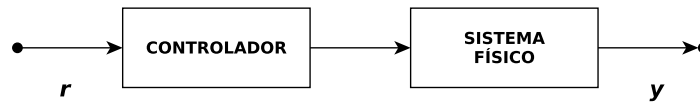


Figura 1. Sistema em malha aberta

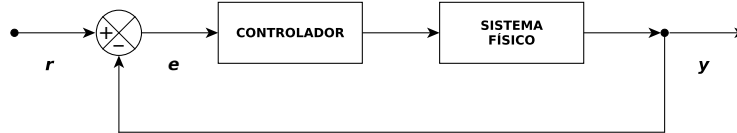


Figura 2. Sistema em malha fechada

Existem métodos para a sua obtenção através da resposta em frequência, apresentados nesta seção a partir do trabalho de [Marlin 2014]. Foram enfatizados aqueles que são utilizados mais frequentemente nas indústrias.

- *Método de Smith* – o método propõe que os valores de θ e de τ sejam selecionados de tal modo que o modelo e as respostas reais coincidam em dois pontos na que a região de elevada taxa de mudança.
- *Método de Sundaresan e Krishnaswamy* – este método também evita a utilização do ponto de inflexão para estimar a constante de tempo τ e de atraso de transporte θ . Eles propuseram que dois tempos, t_1 e t_2 , sejam estimados a partir da curva de resposta a um degrau, correspondente aos 35,3% e 85,3% da resposta, respectivamente.
- *Método de Nishikawa* – o método consiste em determinar os valores das constantes usando o cálculo das áreas.
- *Método Harriot* – Harriot plotou a resposta na forma fracionária de segunda ordem (sem tempo morto) assim mesmo $\frac{t}{t_1+t_2}$ para diversas frações de $\frac{t_2}{t_1}$. Encontrou que todas as curvas se intersectam aproximadamente a 73% do valor final do estado estável onde $\frac{t}{t_1+t_2}$ é igual a 1,3.

Controladores PID

Uma forma de controlador usado amplamente no controle de processos industriais é chamado de três termos ou controlador PID. Este controlador tem uma função de transferência como visto na Equação (5):

$$G(s) = K_p + \frac{K_i}{s} + K_d s \quad (5)$$

onde K_p é o ganho proporcional, K_i é o ganho integral e K_d é o ganho derivativo.

Este controlador oferece controle com três modos: o proporcional, o integral e o derivativo. Cada combinação dos modos possui uma particularidade em sua função de transferência:

- *Controlador proporcional integral* $G(s) = K_p + \frac{K_i}{s}$;
- *Controlador proporcional derivativo* $G(s) = K_p + K_d s$.

Muitos processos industriais são controlados usando controladores PID. Sua popularidade é atribuída parcialmente ao seu bom desempenho em uma ampla faixa de condições de operação e sua simplicidade de funcionamento.

Algoritmos Genéticos

Algoritmos Genéticos (AG) [Goldberg 1989] pertencem a uma classe de heurísticas que usa os conceitos provenientes do princípio de seleção natural para a resolução de uma série de problemas de maximização e minimização. Apresenta uma característica robusta e de fácil adaptação ao cenário desejado. O seu funcionamento foi inspirado na Teoria da Evolução de Darwin que explica sobre a evolução das espécies.

Basicamente, o que um algoritmo genético faz é criar uma população de possíveis respostas e as submete ao processo de evolução obedecendo as seguintes etapas:

- Avaliação: Verifica-se a aptidão das soluções proposta pela população gerada, observando se a mesma presta um bom desempenho.
- Seleção: Os indivíduos são selecionados quanto a sua aptidão através do método de torneio que consiste em sortear os indivíduos e verificar qual dele possui a melhor aptidão.
- Recombinação: Após a seleção os indivíduos passam pelo processo de recombinação de seus alelos (cada elemento do vetor que representa um indivíduo, vetor este chamado cromossomo). É responsável pela convergência da população.
- Mutação: Os alelos podem vir a sofrer modificações. É responsável pela diversidade da população.
- Atualização: Os novos indivíduos gerados são colocados como a população principal.

Método proposto

Indivíduo proposto

A estrutura adotada para o cromossomo teve como base o fato de que o ganho da planta em malha aberta é mais simples de ser determinado pelo método empírico, pois basta realizar a divisão da variação de saída pela variação de entrada. Dessa forma, pode-se considerar que o valor do termo constante do polinômio divisor (constante que não possui multiplicação por nenhum s) é igual a 1. Isso pode ser comprovado usando o teorema do valor final [Marlin 2014] apresentado na Equação (6). Note que, no caso deste trabalho, $F(s)$ é a convolução da função de transferência da planta com uma entrada $u(t)$ em degrau ($\mathcal{L}[u(t)] = s^{-1}$).

$$\lim_{s \rightarrow 0} s \cdot F(s) = \lim_{s \rightarrow 0} s \cdot \left[\frac{1}{s} \frac{K \exp(-\theta s)}{b_n s^n + \dots + b_1 s + 1} \right] = \frac{K}{1} = K \quad (6)$$

Desta forma podemos determinar K como na Equação (7):

$$K = \frac{\Delta y(t)}{\Delta x(t)} = \frac{y(\infty) - y(0)}{x(\infty) - x(0)} \quad (7)$$

onde $y(t)$ é a saída do sistema e $x(t)$ sua entrada.

Por conta das possíveis diferenças nas medidas de um sinal, neste trabalho utiliza-se a média das 5 últimas medições do sinal que captura a variação de saída (já em regime permanente) quando o sistema é submetido a um degrau para o cálculo de K .

Já em malha fechada, o AG fica encarregado de convergir o ganho como os outros parâmetros da função de transferência.

Outra garantia a ser considerada foi o cuidado para que a ordem do divisor não ficasse menor que o dividendo. Um cenário onde isso não seja garantido traz problemas para os sistemas em estudo. Portanto, foram desconsiderados os zeros na montagem do cromossomo. Como o método proposto permite com que o usuário escolha qual a ordem da função de transferência que se aproxima da curva amostrada, temos na Figura 3 a definição do cromossomo adotado. Sempre o termo constante do polinômio $b_0 = 1$.

θ	K	b_n	b_{n-1}	\dots	b_1
----------	-----	-------	-----------	---------	-------

Figura 3. Cromossomo proposto de tamanho igual a $n + 2$; note que em malha aberta K é calculado, deixando de ser alelo.

Na Figura 3, tem-se que: θ é o atraso da planta; K é o ganho (em malha aberta, K é calculado); n é a ordem; e b_i com $1 \leq i \leq n$ são os coeficientes do polinômio do denominador da função de transferência da planta. Um cromossomo destinado a encontrar funções de 1ª ordem tem 3 alelos, um de 3ª ordem possui 5 alelos, e assim por diante. Logo, seu tamanho é determinado por $n + 2$.

Operadores genéticos utilizados

Ambos os operadores, tanto o de recombinação quanto o de mutação, ao serem aplicados, testam alelo por alelo do cromossomo a fim de operar ou não os mesmos. Esses testes correm com a probabilidade igual ao inverso do tamanho do cromossomo. Em média, um alelo será modificado, sendo possível a modificação de todos eles.

Recombinação (*crossover*) – para o trabalho foi adotado o método de recombinação *Simulated Binary Crossover* (SBX) [Deb 1995], aqui descrito pelas Equações (8) e (9).

$$\beta_{q_i} = \begin{cases} (2 u_i)^{\frac{1}{\eta+1}}, & \text{se } u_i \leq 0,5; \\ \left(\frac{1}{2(1-u_i)} \right)^{\frac{1}{\eta+1}}, & \text{caso contrário.} \end{cases} \quad (8)$$

$$\begin{aligned} x_i^{(1,t+1)} &= 0,5 (1 + \beta_{q_i}) x_i^{(1;t)} + 0,5 (1 - \beta_{q_i}) x_i^{(2;t)}, \\ x_i^{(2,t+1)} &= 0,5 (1 - \beta_{q_i}) x_i^{(1;t)} + 0,5 (1 + \beta_{q_i}) x_i^{(2;t)}. \end{aligned} \quad (9)$$

onde o índice i é uma referência ao i -ésimo alelo; β_{q_i} é definido pela Equação (8); $x_i^{(p;q)}$ é o valor do i -ésimo alelo para o p -ésimo filho na q -ésimo geração; $u_i = \mathcal{U}(0,1)$, uma variável aleatória com distribuição uniforme entre 0 e 1. Para esse trabalho, $\eta = 3$.

Note que, como não pode existir coeficientes negativos, existe um teste que traz para zero os alelos que porventura se tornem negativos ao final da recombinação.

Mutação – para o trabalho foi adotado o método de mutação de troca de valores por alelo. O novo valor assumido é aleatoriamente atribuído com auxílio de sorteio com distribuição uniforme entre os limites considerados neste trabalho¹ (entre 0 e 40).

Avaliação proposta

Este é um problema de minimização do erro quadrático médio entre a curva de saída estimada pela função de transferência codificada por um dado indivíduo e a curva de referência amostrada na saída da planta quando submetida a uma entrada em degrau. Como o método proposto foi implementado em MATLAB®, as funções de avaliação aqui apresentadas estão nessa linguagem.

Note que para o cálculo do erro são necessários a curva de referência amostrada, o tempo amostrado e o ganho do degrau aplicado à planta para a captura de sua resposta.

A avaliação (*fitness*) do indivíduo segue o seguinte algoritmo quando se tratar de malha aberta:

```
% Código MATLAB
function f = fitness(indiv, referencia, tempo, ganho_degrau)
    K = mean(referencia(end-4:end))/ganho_degrau;
    H = tf(K, [indiv(3:end) 1]);
    H.OutputDelay = indiv(1);
    estimado = step(H, tempo);
    f = sqrt(sum((estimado-referencia).^2));
end
```

E segue o seguinte algoritmo, considerando a convolução do controlador com a planta e a retroalimentação, quando se tratar de malha fechada:

```
% Código MATLAB
function f = fitness(indiv, referencia, tempo, ganho_degrau, PID_ref)
    H = tf(indiv(2)*ganho_degrau, [indiv(3:end) 1]);
    H.OutputDelay = indiv(1);
    H = ss(H*pid(PID_ref));
    estimado = step(H, tempo);
    f = sqrt(sum((estimado-referencia).^2));
end
```

Parâmetros do AG proposto

O AG foi configurado com valores em torno dos considerados padrão na literatura:

- População: 200
- Gerações: 50 (malha aberta) e 200 (malha fechada)
- Número de indivíduos no torneio: 3
- Probabilidade de recombinação: 0,95 (na falha, os pais selecionados são copiados para a próxima geração)
- Probabilidade de mutação: 0,15
- Atualização: substituição dos pais pelos filhos

¹Aqui foi considerado que os coeficientes da função de transferência da planta não podem ser negativos e que um máximo de 40 é coerente com a prática na área.

Considerações sobre o método

Como o usuário pode determinar a ordem com a qual deseja trabalhar para estimar a função de transferência de uma planta a partir de sua resposta ao degrau amostrada, existe a possibilidade de se usar uma ordem superior a da realidade. Se for esse o caso, o próprio AG irá convergir para soluções onde os coeficientes de maior grau sejam iguais a zero. Com isso, pode-se afirmar que o AG irá reduzir a ordem da função estimada se for necessário.

Resultados

Planta Fictícia em Malha Aberta

Para o experimento foi adotado uma planta fictícia de 3ª ordem com atraso para a qual são usados os métodos convencionais para se determinar a função de transferência e confrontar com o resultado encontrado pelo método proposto. A Figura 4 representa sua resposta ao degrau unitário, gerada com o auxílio de sua função de transferência na Equação (10). De posse da curva gerada, a função de transferência não é utilizada no método, sendo usada apenas para comparar se a resposta obtida condiz com a esperada.

$$F(s) = \frac{0,5 \exp(-10 s)}{2 s^3 + 4 s^2 + 5 s + 1} \quad (10)$$

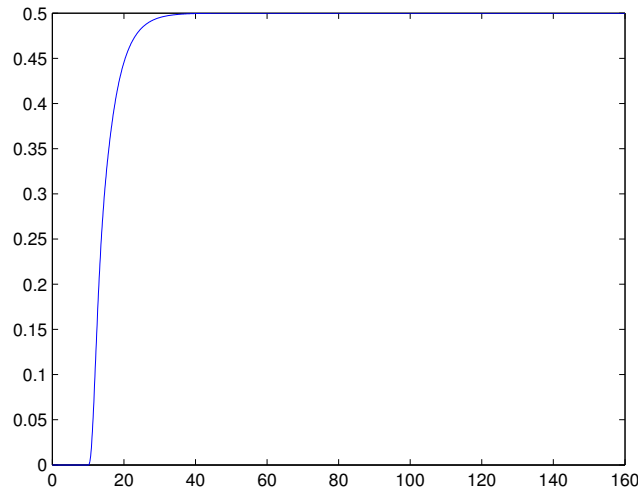


Figura 4. Planta fictícia, malha aberta

O sistema foi modelado com estimativas de 1ª, de 2ª e de 3ª ordens utilizando diversos métodos conhecidos. Nem todos os métodos são aplicáveis para 2ª ordem e somente o método proposto pôde ser aplicado para 3ª ordem.

Foram validadas as funções de transferência através do erro quadrático para a entrada de um degrau unitário. As Tabelas 1, 2 e 3 mostram os resultados em malha aberta obtidos nesta etapa. Na primeira coluna, é identificado cada método utilizado; na segunda coluna, o modelo obtido; na terceira coluna, o erro quadrático quando comparado com a curva de referência. No caso dos métodos apresentados na Seção 3.3, o modelo

resultante é obtido de maneira determinística. No caso do específico do AG proposto, os modelos apresentados nessas tabelas assumem como parâmetros os parâmetros médios alcançados em 30 execuções — exceto o ganho que é previamente calculado usando a Equação (7) —, sendo que na coluna do erro, μ identifica a média e σ identifica o desvio padrão do erro de aproximação da curva de referência.

Tabela 1. Modelagem do sistema fictício para primeira ordem para malha aberta.

Método	Modelo encontrado	Erro
AG proposto	$F_{médio}(s) = \frac{0,5}{4,06s + 1} e^{-10,94s}$	$\mu = 0,045$ $\sigma = 3,63 \cdot 10^{-7}$
Smith	$F(s) = \frac{0,5}{3,855s + 1} e^{-11,1s}$	0,233
Sundaresan e Krishnaswamy	$F(s) = \frac{0,5}{4,04s + 1} e^{-11,1s}$	0,325
Nishikawa	$F(s) = \frac{0,5}{4,07s + 1} e^{-10,9s}$	0.144

Note que foram utilizados dois métodos de modelagem para segunda ordem. Como esses métodos não fazem o cálculo do atraso, o mesmo foi calculado a parte e depois adicionado aos modelos encontrados.

Tabela 2. Modelagem do sistema fictício para segunda ordem para malha aberta.

Método	Modelo encontrado	Erro
AG proposto	$F_{médio}(s) = \frac{0,5}{1,93s^2 + 4,49s + 1} e^{-10,49s}$	$\mu = 0,041$ $\sigma = 0,016$
Harriot	$F(s) = \frac{0,5}{3,686s^2 + 4,8s + 1} e^{-10,0s}$	0,310
Smith	$F(s) = \frac{0,5}{5,244s^2 + 4,58s + 1} e^{-10,0s}$	0,587

Os autores não encontraram métodos analíticos que pudessem modelar sistemas de 3ª ordem. Desta forma, foi utilizado somente o algoritmo proposto como modelador.

Tabela 3. Modelagem do sistema fictício para terceira ordem para malha aberta.

Método	Modelo encontrado	Erro
AG proposto	$F_{médio}(s) = \frac{0,5}{2,965s^3 + 4,826s^2 + 5,184s + 1} e^{-9,82s}$	$\mu = 0,023$ $\sigma = 0,024$

Planta Fictícia em Malha Fechada

Existem plantas industriais na realidade para as quais existem uma série de dificuldades de se interromper o processo e colocá-las em malha aberta. Por exemplo, o controle de temperatura de petróleo. Somada à dificuldade matemática para a modelagem da mesma, optou-se também em adaptar o método proposto para determinar o modelo matemático da função de transferência em malha fechada. A estrutura do algoritmo usado foi a análoga a da usada na malha aberta, alterando-se apenas parte da função de avaliação. Note que em malha fechada não é possível predefinir o valor do ganho K da planta a ser modelada, uma vez que a mesma é ligada em série com o compensador e perde essa referência. Em outras palavras, o parâmetro do ganho integra o cromossomo para também ser estimado pelo AG.

Em ambientes industriais onde se desconhece a função de transferência de uma planta que deseja-se controlar, é comum o uso do método de Ziegler e Nichols [Marlin 2014]. Assim, com o uso deste método determina-se os parâmetros do controlador PID apresentado na Equação (11).

$$\text{PID} = 1,215 + \frac{0,0422}{s} \quad (11)$$

Com o uso do MATLAB®, código apresentado a seguir, foi gerada a curva de referência (Figura 5) para o AG. Note que não foi conseguido um controle suave.

```
g = tf([0.5],[2 4 5 1]);
g.OutputDelay = 10;
g1 = ss(pid(1.215,0.0422,0)*g);
h = feedback(g1,1);
[referencia,tempo] = step(h,0:0.1:450);
ganho_degrau = 1;
```

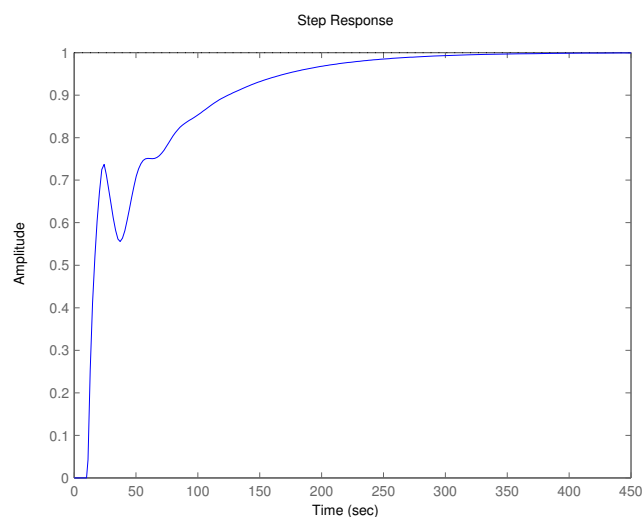


Figura 5. Planta fictícia em malha fechada com controlador PID da Equação (11)

Aplicando o algoritmo modificado para determinação em malha fechada, foram obtidos os resultados das Tabelas 4 e 5 para estimação de primeira, segunda e terceira

ordem da planta fictícia descrita pela Equação 10 a partir de uma curva de referência amostrada em situação de malha fechada com conhecimento prévio dos parâmetros do conversor PID atuante.

Tabela 4. Modelagem por ordem do sistema fictício em malha fechada (DP é o desvio-padrão).

Parâmetro	1ª ordem		2ª ordem		3ª ordem	
	Média	DP	Média	DP	Média	DP
θ	10,80	0,343	9,988	0,269	9,951	0,307
K	0,501	$1,949 \cdot 10^{-3}$	0,500	$0,103 \cdot 10^{-3}$	0,500	$0,513 \cdot 10^{-3}$
coeficientes	4,205	0,370	4,018	$69,80 \cdot 10^{-3}$	1,999	0,134
			4,984	$6,092 \cdot 10^{-3}$	4,004	0,133
					5,000	0,011
Erro	$14,64 \cdot 10^{-3}$	$14,72 \cdot 10^{-3}$	$7,735 \cdot 10^{-3}$	$3,384 \cdot 10^{-4}$	$2,344 \cdot 10^{-3}$	$2,696 \cdot 10^{-3}$

Tabela 5. Modelos estimados para a planta da Equação (10) em malha fechada.

Ordem	Modelo Estimado
1ª	$F_{\text{médio}}(s) = \frac{0,501}{4,205s + 1} e^{-10,80s}$
2ª	$F_{\text{médio}}(s) = \frac{0,500}{4,018s^2 + 4,984s + 1} e^{-9,988s}$
3ª	$F_{\text{médio}}(s) = \frac{0,500}{1,999s^3 + 4,004s^2 + 5,000s + 1} e^{-9,951s}$

Como pode-se observar no modelo de 3ª ordem estimado pelo AG e descrito na Tabela 5, na comparação com o modelo da planta fictícia descrita na Equação (10), a estimação foi praticamente certa. Note que os autores não encontraram na literatura métodos que pudessem ser usados no caso de malha fechada, o que inviabilizou comparações com o AG.

Conclusão

Através de todas as simulações e validações realizadas, demonstra-se que o Algoritmo Genético proposto atende as expectativas de ser uma ferramenta de modelagem de plantas industriais com funções de transferência desconhecidas.

Na primeira etapa, a função de transferência desconhecida em malha aberta foi aproximada para um sistema de primeira, segunda e terceira ordens. Os métodos que apresentaram maior aproximação foram a do AG proposto e o de Nishikawa, para primeira ordem. O AG proposto se destacou com um erro quadrático inferior aos dois métodos convencionais disponíveis para segunda ordem (note que os métodos convencionais somente conseguem determinar a função de transferência sem atraso e que foi necessário

adicioná-lo após a aplicação do respectivo método). Para uma aproximação de terceira ordem, não houve comparação com nenhum outro método, pois não foram encontrados métodos para aproximação em tal cenário.

Na segunda etapa, existe a possibilidade de malha fechada com um controle nada suave. O AG proposto também se mostrou robusto, uma vez que em todas as ordens de aproximação, convergiu praticamente para a mesma função que havia convergido em malha aberta. Ainda, a aproximação de terceira ordem foi praticamente a função de transferência desejada.

Em vista dos resultados apresentados é válido afirmar que o AG aqui proposto é uma importante ferramenta de modelagem, seja em malha aberta ou fechada, e que ainda é viável para aplicações industriais uma vez que torna-se viável para processos ininterruptos onde não é possível abrir a malha de controle. Como trabalhos futuros, é planejado a aplicação deste método em plantas industriais reais para observarmos a robustez do método proposto.

Referências

- Biradar, S., Hote, Y. V., and Saxena, S. (2016). Reduced-order modeling of linear time invariant systems using big bang big crunch optimization and time moment matching method. *Applied Mathematical Modelling*, 40(15–16):7225 – 7244.
- Cabral, H. A. and de Melo, M. T. (2011). Using genetic algorithms for device modeling. *IEEE Transactions on Magnetics*, 47(5).
- Deb, Kalyanmoy; Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148.
- Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Professional.
- He, T., Hong, L., Kaufman, A., and Pfister, H. (1996). Generation of transfer functions with stochastic search techniques. In *Proceedings of the 7th Conference on Visualization '96, VIS '96*, pages 227–ff., Los Alamitos, CA, USA. IEEE Computer Society Press.
- Loyka, S. L. (1999). Applying genetic algorithm to modeling nonlinear transfer functions. In *Telecommunications in Modern Satellite, Cable and Broadcasting Services, 1999. 4th International Conference on*, volume 1, pages 247–250 vol.1.
- Marlin, T. E. (2014). *Process Control: designing processes and control systems for dynamic performance*. McGraw-Hill, Inc., 2 edition.
- McCall, J. (2005). Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics*, 184:205–222.
- Nedelea, L., Neag, M., Topa, M., Festila, L., and Topa, V. (2006). Analog filter transfer function optimization based on a genetic algorithm. In *Applied Electronics*, Pielsen.
- Tavakolpour, A. R., Darus, I. Z. M., Tokhi, O., and Mailah, M. (2010). Genetic algorithm-based identification of transfer function parameters for a rectangular flexible plate system. *Engineering Applications of Artificial Intelligence*.

Classificação de *tweets* por relevância para concepção de um modelo de aprendizado de máquina para uso em aplicações de TV Social

Cássio Faria da Silva¹, Cesar Augusto Camillo Teixeira¹, Helena de Medeiros Caseli¹

¹Departamento de Computação – Universidade Federal de São Carlos (UFSCar)
Rodovia Washington Luís, km 235 – 13565-905 – São Carlos – SP – Brasil

cassiofs@gmail.com, {cesar,helenacaseli}@dc.ufscar.br

Resumo. *Apresentam-se resultados de pesquisa, realizada com a classificação automática de tweets, com o objetivo de selecionar mensagens postadas no Twitter que tenham potencial de incentivar a participação de telespectadores em aplicativos voltados a TV Social. Com o intuito de alcançar o objetivo proposto, foram aplicados os algoritmos Máquina de Vetores de Suporte (ou Support Vector Machine, SVM) e Naive Bayes na classificação de um corpus de tweets postados usando a hashtag #masterchefbr, elaborado especificamente para esta finalidade. Os resultados obtidos com os classificadores propostos demonstraram que o SVM apresentou um melhor resultado com precisão de 87,1%.*

1. Introdução

Segundo [Klym e Montpetit 2008], o termo TV Social é definido como a conjunção de serviços de vídeo que integram outros serviços de comunicação como voz, chat, recomendação, sensibilidade ao contexto, classificação e interatividade pelos pares para apoiar uma experiência de TV compartilhada. A experiência de assistir TV não necessariamente deve ser realizada de forma solitária e a discussão social pode melhorar essa experiência.

O consumo de conteúdo televisivo está se tornando cada vez mais social e interativo. As redes sociais mais populares, como Twitter¹ e Facebook², estão sendo muito utilizadas para o propósito de interação entre telespectadores [Cesar e Geerts 2011].

Evidências indicam que, embora o Twitter possa satisfazer algumas das necessidades básicas dos telespectadores de comentar e compartilhar a experiência de assistir TV, as limitações desse tipo de serviço, em determinados cenários, podem restringir a interação e que é comum o telespectador perder o interesse de interagir em pouco tempo [Wang et al. 2014, Pagani e Mirabello 2011]. Por não ter sido o Twitter desenvolvido especificamente com o objetivo de promover interação entre espectadores de televisão, abre-se um potencial inexplorado no fomento da interação social entre telespectadores. Assim, o desenvolvimento de recursos e ferramentas adequadas voltadas para telespectadores pode melhorar a experiência de assistir TV.

Ferramentas desenvolvidas para incentivar discussões entre telespectadores, projetadas especificamente com esse propósito, levando-se em conta as informações contex-

¹<http://www.twitter.com>

²<http://www.facebook.com>

tuais do momento e pertinentes à programação de TV, podem aumentar a quantidade de interações e melhorar a qualidade da experiência em assistir TV.

Algoritmos de classificação fundamentados em algoritmos de aprendizado de máquina [Mitchell 1997] já são bastante utilizados na criação de filtros de texto. Para a classificação de *tweets*, algoritmos de aprendizado de máquina são aplicados no problema de classificação de prioridade de *tweets*, segundo conceitos individuais de relevância e importância.

Neste trabalho propõe-se um modelo de filtro de mensagens relevantes, baseado em métodos de aprendizado de máquina, com o objetivo de utilizá-lo em uma aplicação de TV Social. O Twitter foi escolhido como base para a composição do corpus pois o formato de mensagens da aplicação proposta para TV Social se assemelha ao formato de suas mensagens. Neste trabalho são apresentados os resultados obtidos com a aplicação dos algoritmos Máquina de Vetores de Suporte (ou *Support Vector Machine*, SVM) e Naive Bayes na classificação de *tweets* por relevância postados no dia 2 de maio de 2017, com a *hashtag* #masterchefbr durante a exibição do nono episódio do MasterChef Brasil.

O MasterChef Brasil, *talent show* de culinária transmitido pela Rede Bandeirantes, foi escolhido como objeto de estudo nesta pesquisa devido a sua audiência e participação no Twitter. De acordo com dados apurados pelo Kantar Twitter TV Ratings³, levantamento que mensura a repercussão do conteúdo televisivo no ambiente digital, o MasterChef Brasil foi o programa de televisão com a maior participação na semana de 1 a 5 de maio de 2017, considerando-se o número total de *tweets*: 107,8 mil *tweets* durante a exibição do nono episódio da quarta temporada do programa. Desse modo, para a realização dos experimentos descritos neste artigo, um corpus formado por 450 *tweets* usando a *hashtag* #masterchefbr foi construído.

O restante deste trabalho está organizado da seguinte forma. A seção 2 cita alguns trabalhos que realizam classificação de *tweets* utilizando aprendizado de máquina. Na seção 3 é descrito o processo de coleta e anotação do corpus utilizado nos experimentos. A seção 4 relata os experimentos realizados e os resultados obtidos com a aplicação dos algoritmos investigados. As considerações finais, com a análise dos resultados alcançados e os trabalhos futuros são apresentados na seção 5.

2. Trabalhos Relacionados

Diversos trabalhos da literatura já aplicaram algoritmos de aprendizado de máquina com o propósito de analisar e classificar mensagens e de criar filtros automáticos de conteúdos textuais. Com esse objetivo, abordagens supervisionadas e não supervisionadas têm sido empregadas em serviços de correio eletrônico e redes sociais virtuais. Assim, procurou-se pesquisar na literatura alguns trabalhos referenciais que indicassem semelhanças com o objetivo deste trabalho.

Dentre estes, destacam-se os seguintes. Em [Cobo et al. 2015], os autores apresentam um classificador com o objetivo de classificar os *tweets* em relevante ou não relevante para comunicar sobre um terremoto. Um corpus de 2187 *tweets* escritos em espanhol foi coletado para ser testado em experimentos realizados com 5 algoritmos: Baseline,

³<https://www.kantaribopemedia.com/kttr-type/kantar-twitter-tv-ratings-0105-a-070517/>. Acesso em: 14 maio 2017.

Bernoulli NB, Logistic Regression, Linear SVM e Random Forest. Como resultado, os autores relatam uma precisão de 83,1% com o Bernoulli NB e 67,7% com Linear SVM.

Em [Tsgawa et al. 2015], os autores avaliaram a eficácia de usar as atividades de mídia social de um usuário para estimar o seu grau de depressão. Através de experimentos usando o algoritmo SVM foi demonstrado que os dados obtidos das atividades dos usuários no Twitter podem ser usados para prever a depressão desses usuários com uma precisão de, aproximadamente, 69%. Os dados usados nos experimentos foram obtidos por meio de um questionário aplicado a voluntários de língua japonesa e da atividade do usuário na rede social.

Uma arquitetura para classificação de usuários do Twitter é apresentada em [Pennacchiotti e Popescu 2011]. Os autores utilizaram o algoritmo de classificação *Gradient Boosted Decision Trees* (GBDT) para algumas tarefas. Por exemplo, na previsão de afiliação política, os autores relatam uma precisão de 98,9% e 92% na classificação de usuários como democratas e republicanos, respectivamente.

Com o objetivo de analisar spam e confiabilidade das avaliações de produtos da amazon.com, [Jindal e Liu 2008] analisa essas atividades e apresenta técnicas para detectá-las. A base de dados foi composta por 5,8 milhões de críticas e 2,14 milhões de revisores. Para a detecção de spam foi usada regressão logística, já que SVM e Naive Bayes não apresentaram bom desempenho. [Aberdeen et al. 2010] também apresenta um modelo estatístico (regressão logística) propondo o recurso *Priority Inbox* do Gmail, com o objetivo de classificar o e-mail pela probabilidade do usuário executar uma ação nesse e-mail.

SVM e Naive Bayes também foram usados para análise de sentimentos em [Pang et al. 2002]. Nesse trabalho, o foco estava na determinação do sentimento geral de uma revisão como positiva ou negativa num corpus de 2053 avaliações de filmes (752 negativas e 1301 positivas). Para tanto, foram usados os algoritmos SVM, Naive Bayes e Máxima Entropia sendo que SVM apresentou um melhor desempenho.

Outros trabalhos que realizaram a análise de sentimentos a partir de *tweets*, desta vez escritos em português, foram [Souza e Vieira 2012] e [Alves et al. 2014]. Em [Souza e Vieira 2012], foi feita a análise de sentimentos em um corpus de 1700 *tweets* escritos em português. Enquanto que em [Alves et al. 2014], foi apresentado um estudo de caso para análise de sentimentos utilizando SVM e Naive Bayes, em um corpus de 1500 *tweets* escritos em português e produzidos durante a Copa das Confederações da FIFA 2013. Esse *tweets* foram rotulados manualmente por 10 anotadores. Os resultados obtidos indicaram que o algoritmo SVM, com uma acurácia de 87%, superou o Naive Bayes, que obteve uma acurácia de 72%.

3. Construção do Corpus

O MasterChef Brasil é um *talent show* de culinária brasileiro exibido pela Rede Bandeirantes, baseado no consagrado formato original de mesmo nome exibido pela BBC (*British Broadcasting Corporation*) no Reino Unido. O programa é apresentado pela jornalista Ana Paula Padrão e os jurados são os chefs de cozinha Henrique Fogaça, Erick Jacquin e Paola Carosella.

A primeira temporada estreou em 2 de setembro de 2014 e terminou no dia 16 de

dezembro do mesmo ano. A segunda temporada estreou em 19 de maio de 2015, dessa vez com 18 participantes, dois a mais que a anterior. A terceira temporada estreou em 15 de março de 2016, com 21 participantes, três a mais que a edição anterior. A quarta temporada estreou em 7 de março de 2017, com 21 participantes, mesmo número que a edição anterior.

O MasterChef Brasil foi escolhido como objeto de estudo desta pesquisa devido a sua audiência e participação no Twitter. De acordo com dados apurados pelo Kantar Twitter TV Ratings⁴, o *talent show* foi o programa de televisão com a maior participação na semana de 1 a 5 de maio de 2017, considerando-se o número total de *tweets*: 107,8 mil *tweets* durante a exibição do nono episódio da quarta temporada do programa.

Para a captura dos *tweets*, foi utilizada a ferramenta TAGS⁵, modelo de planilha do Google que permite executar uma coleta automatizada de resultados de pesquisa do Twitter. Utilizando essa ferramenta, foram capturados aproximadamente 48 mil *tweets* durante a exibição do programa, entre 22:36 e 00:56 do dia 2 de maio de 2017 (nono episódio). Cada *tweet* é composto por 17 campos, como descrito na Tabela 1: (1) *id_str*, (2) *from_user*, (3) *text*, (4) *created_at*, (5) *geo_coordinates*, (6) *user_lang*, (7) *in_reply_to_user_id_str*, (8) *in_reply_to_screen_name*, (9) *from_user_id_str*, (10) *in_reply_to_status_id_str*, (11) *source*, (12) *profile_image_url*, (13) *user_followers_count*, (14) *user_friends_count*, (15) *user_location*, (16) *status_url*, (17) *entities_str*.

Os *tweets* capturados foram, então, selecionados para compor o corpus de 450 *tweets* descrito neste artigo. Para tanto, o principal critério utilizado na seleção dos *tweets* foi o número de seguidores do usuário que produziu o *tweet*, ou seja, os *tweets* capturados foram ordenados decrescentemente pelo campo *user_followers_count* e os primeiros colocados foram selecionados para compor o corpus final. Além desse critério de ordenação, também optou-se por selecionar *tweets* postados em momentos importantes do nono episódio do programa MasterChef Brasil 2017, o qual foi composto, basicamente, por cinco momentos:

- 1. Prova em grupo** – momento em que os participantes foram divididos em dois grupos e os membros de cada grupo elaboraram, em conjunto, um prato especificado pelos organizadores do programa.
- 2. Avaliação da prova em grupo** – momento em que os pratos da prova em grupo foram avaliados e julgados. Os membros do grupo de melhor desempenho nessa prova ficaram imunes à eliminação, enquanto os participantes pertencentes ao outro grupo foram para uma prova de eliminação.
- 3. Prova de eliminação** – nesse momento, cada membro do grupo derrotado na prova em grupo preparou, individualmente, um prato proposto pelo programa. Aproximadamente 18 mil *tweets* foram capturados entre 11:39 e 00:01 e, dentre eles, selecionados 150 *tweets* produzidos por aqueles usuários com o maior número de seguidores (de acordo com o campo *user_followers_count*).
- 4. Avaliação da prova de eliminação** – nesse momento, os pratos preparados na prova de eliminação foram avaliados e julgados de acordo com os critérios do programa.

⁴<https://www.kantaribopemedia.com/kttr-type/kantar-twitter-tv-ratings-0105-a-070517/> acesso em: 14 maio 2017.

⁵<https://tags.hawksey.info/>. Acesso em: 15 maio 2017.

Tabela 1. Descrição dos campos coletados

Campo	Descrição
<i>id_str</i>	Código que identifica o <i>tweet</i> .
<i>from_user</i>	Usuário que postou o <i>tweet</i> .
<i>text</i>	Texto da mensagem.
<i>created_at</i>	Hora (UTC) que o <i>tweet</i> foi criado.
<i>geo_coordinates</i>	Coordenadas geográficas do local do <i>tweet</i> .
<i>user_lang</i>	Indica o idioma do usuário.
<i>in_reply_to_user_id_str</i>	Se o <i>tweet</i> for resposta de um outro, indica o Id do autor do <i>tweet</i> original.
<i>in_reply_to_screen_name</i>	Se o <i>tweet</i> for uma resposta, indica o nome do autor do <i>tweet</i> original.
<i>from_user_id_str</i>	Id do usuário que postou o <i>tweet</i> .
<i>in_reply_to_status_id_str</i>	Se o <i>tweet</i> for uma resposta, este campo conterá o Id to <i>tweet</i> original.
<i>source</i>	Indica o programa usado para postar o <i>tweet</i> .
<i>profile_image_url</i>	Url da imagem do perfil do usuário.
<i>user_followers_count</i>	O número de seguidores que a conta do usuário possui.
<i>user_friends_count</i>	O número de usuários que esta conta está seguindo.
<i>user_location</i>	A localização definida pelo usuário para o seu perfil.
<i>status_url</i>	Url do <i>tweet</i> .
<i>entities_str</i>	Conjunto de metadados e informações contextuais adicionais sobre o conteúdo do <i>tweet</i> postado.

Aproximadamente 12 mil *tweets* foram capturados entre 00:02 e 00:29 e, dentre eles, selecionados 150 *tweets* de usuários com o maior número de seguidores (de acordo com o campo *user_followers_count*).

5. Eliminação – momento em que o resultado da avaliação é revelado e o participante com pior desempenho é eliminado e se despede do programa. Aproximadamente 18 mil *tweets* foram capturados entre 00:31 e 00:55 e, dentre eles, selecionados 150 *tweets* de usuários com o maior número de seguidores (de acordo com o campo *user_followers_count*).

Assim, dos 48 mil *tweets* capturados inicialmente, foram selecionados 150 mensagens postadas pelos usuários em cada um dos três momentos mais importantes do programa – Prova de eliminação, Avaliação da prova de eliminação e Eliminação –, totalizando 450 *tweets*.

Após o processo de coleta do corpus deu-se início ao processo de anotação manual dos *tweets* em uma das duas classes: (1) relevante e (2) não relevante. No contexto desta pesquisa, considera-se como relevante uma mensagem que seja capaz de estimular a interação e incentivar outros telespectadores a postar suas mensagens. Para tentar diminuir a subjetividade inerente a essa tarefa, foram definidos alguns critérios para anotação de um *tweet* como relevante ou não, como apresentado a seguir.

***tweet* rotulado como “relevante” (S) :**

- *Tweet* com texto formulado no sentido de expressar uma torcida a um participante;

- *Tweet* cujo texto representa uma pergunta referente ao contexto do programa;
- *Tweet* com texto bem-humorado, que sugestione a criação de memes⁶ de internet;
- *Tweet* que representa uma situação ou momento de destaque no contexto do programa.

***tweet* rotulado como “não relevante” (N) :**

- *Tweet* contendo somente uma palavra;
- *Tweet* em que o usuário expressa uma opinião ou pensamento fora do contexto do momento ou do programa;
- *Tweet* contendo apenas figuras ou links externos;
- *Tweet* escrito em outras línguas.

Assim, o texto (*text*) de cada *tweet* foi rotulado manualmente por dois anotadores, usuários de redes sociais e que acompanham o programa. Vale mencionar que a anotação ideal seria aquela realizada por especialistas influenciadores digitais, o que não foi possível neste primeiro experimento. Entretanto, está em andamento a coleta de um corpus de *tweets* pré-selecionados por influenciadores digitais que poderá ser utilizado, em um trabalho futuro, para validar os resultados obtidos nesta pesquisa.

Após o processo de anotação inicial, em função de alguns conflitos (um mesmo *tweet* recebendo classificação manual diferente), um processo de rotulação adicional, para resolver as divergências encontradas, foi realizado.

Ao final desse processo, tem-se que o corpus é composto por dois atributos: (1) *tweet*, o texto do *tweet* (campo *text*) e (2) *relevante?*, a classe rotulada que indica se o texto é relevante (S) ou não (N). O corpus gerado possui uma distribuição balanceada das classes S e N, com 253 *tweets* (56,3%) rotulados como N e 197 *tweets* (43,7%) rotulados como S.

A Tabela 2 mostra alguns exemplos de *tweets* que compõem o corpus. A tabela exhibe o texto do *tweet* e a classe rotulada indicando se a mensagem é relevante ou não. Os textos 1, 2 e 3 foram rotulados por ambos os anotadores como relevantes, já os textos 4 e 5 foram considerados irrelevantes pelos anotadores.

Tabela 2. Exemplos de *tweets*

	<i>Tweet</i>	Relevante?
1	Gente, maior climão	S
2	programa tá demais hoje	S
3	A cada dia que passa o mezanino se torna mais desejado!	S
4	Finalmente vou aprender a cozinhar um belo peixe :)	N
5	MARAVILHOSO	N

Com o objetivo de normalizar os textos das mensagens e de pré-processar o corpus colocando-o num formato adequado para o uso na ferramenta WEKA⁷, foi desenvolvido

⁶A expressão meme de Internet é usada para descrever um conceito de imagem, vídeo e/ou relacionados ao humor, que se espalha via Internet. O termo é uma referência ao conceito de memes, que se refere a uma teoria ampla de informações culturais criada por Richard Dawkins em 1976 no seu livro *The Selfish Gene*. ([https://pt.wikipedia.org/wiki/Meme_\(Internet\)](https://pt.wikipedia.org/wiki/Meme_(Internet))). Acesso em: 14 jun. 2017.)

⁷<http://www.cs.waikato.ac.nz/ml/weka/>

um *script* em linguagem Python⁸ para: (i) eliminar acentos, caracteres especiais e links para outros sites, (ii) retirar espaços e caracteres duplicados, (iii) padronizar os nomes dos participantes e (iv) remover *stop-words*. Além destas normalizações no corpus, o *script* ainda prepara o arquivo com a formatação padrão do WEKA.

Após o processo de normalização, o filtro *StringToWordVector* do WEKA foi utilizado para transformar o texto de cada uma das instâncias de treinamento em um conjunto de *tokens*. A execução desse filtro torna possível a aplicação dos algoritmos de aprendizado de máquina disponíveis no WEKA. As configurações padrão do filtro não foram alteradas, pois a normalização do texto, executada no passo anterior, satisfaz todas as necessidades de ajustes de pré-processamento.

O *script* em Python usado para o pré-processamento do corpus, o corpus com os *tweets* anotados como relevantes ou não, e demais recursos produzidos neste trabalho estão disponíveis em <https://sites.google.com/view/classificacaotweetsmasterchef/>.

4. Experimentos e Resultados

Utilizando o WEKA, foram realizados experimentos com o conjunto de 450 *tweets* previamente rotulados e pré-processados. Dois algoritmos de aprendizado de máquina supervisionado foram selecionados para esses experimentos: SVM e Naive Bayes.

O SVM é um classificador binário com bom desempenho em classificação de textos [Pang et al. 2002, Alves et al. 2014] que, em sua essência, constrói um hiperplano ótimo de modo que ele possa separar diferentes classes de dados com a maior margem possível. Para problemas de classificação não lineares, o SVM utiliza uma função kernel cujo objetivo é o de mapear os dados de entrada no espaço original para o espaço de características, que é um espaço de dimensão superior onde o conjunto de treinamento se torna linearmente separável.

O classificador Naive Bayes é um algoritmo do paradigma probabilístico, ou seja, traz a classificação mais provável com base na aplicação do teorema de Bayes com fortes (ingênuas) hipóteses de independência entre seus atributos. Ele foi selecionado por ser um algoritmo simples mas que apresenta bons resultados na classificação de textos.

Para treinar e testar os classificadores SVM e Naive Bayes foi utilizado o WEKA com método de validação cruzada com 10 partições (*10-fold cross-validation*) e parâmetros padrão da ferramenta. As medidas de avaliação utilizadas no experimento foram acurácia, precisão, cobertura e medida F. Essas medidas são definidas nas equações (1), (2), (3) e (4), respectivamente.

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precisão = \frac{TP}{TP + FP} \quad (2)$$

$$Cobertura = \frac{TP}{TP + FN} \quad (3)$$

⁸<https://www.python.org/>

$$Medida F = 2 \times \frac{Precisão \times Cobertura}{Precisão + Cobertura} \quad (4)$$

Nessas equações, TP indica verdadeiro positivo, que é definido como o número de *tweets* que o algoritmo identificou corretamente como “relevante”. TN indica verdadeiro negativo, que é definido como o número de *tweets* que o algoritmo identificou corretamente como “não relevante”. FP indica falso positivo, definido como o número de *tweets* que o algoritmo classificou incorretamente como “relevante”, e FN, falso negativo, que indica o número de *tweets* que o algoritmo classificou incorretamente como “não relevante”.

Considerando-se a medida de acurácia, que avalia quanto o algoritmo acertou em suas tentativas, obteve-se 86,9% de acurácia usando o SVM, ou seja, 86,9% das instâncias foram classificadas corretamente pelo SVM. Os resultados obtidos para as demais medidas são apresentados na Tabela 3.

Tabela 3. Resultados obtidos com SVM em %

Relevante	Precisão	Cobertura	Medida F
S	89,7	79,2	84,1
N	85,1	92,9	88,8
Média Ponderada	87,1	86,9	86,8

O corpus utilizado no experimento também foi testado com o algoritmo Naive Bayes, o qual obteve uma acurácia de 80,6%, ou seja, 80,6% das instâncias foram classificadas corretamente por esse algoritmo. Os resultados obtidos para as outras medidas podem ser analisados na Tabela 4.

Tabela 4. Resultados obtidos com Naive Bayes em %

Relevante	Precisão	Cobertura	Medida F
S	79,6	75,1	77,3
N	81,4	85,0	83,2
Média Ponderada	80,6	80,7	80,6

A comparação dos resultados obtidos com os classificadores SVM e Naive Bayes, considerando-se as medidas apresentadas, demonstra que o SVM apresentou um melhor resultado.

Vale mencionar, também, que esses resultados se aproximam bastante dos apresentados em [Alves et al. 2014] para análise de sentimentos utilizando SVM e Naive Bayes em um corpus de 1500 *tweets* escritos em português produzidos durante a Copa das Confederações da FIFA 2013. Nesse trabalho relacionado, o SVM também foi o algoritmo de melhor desempenho, atingindo uma acurácia de 87%, enquanto a do Naive Bayes foi de 72%.

Além da acurácia, outras três medidas foram analisadas: precisão, cobertura e medida F. Ao analisar os valores obtidos para a precisão, de acordo com a Tabela 3, nota-se que as proporções de S e N estão relativamente balanceadas, o que significa que o algoritmo apresentou uma boa taxa acerto para resultados relevantes (89,7%).

Já os valores para a cobertura foram os que apresentaram o maior desequilíbrio entre as classes S e N, de acordo com os dados da Tabela 3: 79,2% e 92,9%, respectivamente. Esses dados indicam uma grande quantidade de instâncias positivas classificadas como negativas, ou seja, um número elevado de *tweets* que foram avaliados como “relevantes” pelos anotadores, mas como “não relevantes” pelo algoritmo SVM. Contudo, considerando-se a aplicação pretendida (filtro de *tweets* relevantes), vale ressaltar que uma boa precisão para a classe relevante (S) é mais importante do que uma boa cobertura. Isso indica um filtro mais rigoroso de *tweets* que são exibidos para o influenciador digital: mesmo que alguns *tweets* relevantes sejam perdidos (dada a cobertura de 79,2%), os que são mostrados para o influenciador digital têm grandes chances de serem relevantes (dada a precisão de 89,7%).

Já a medida F, que se refere à média harmônica de precisão e cobertura, também apontou um relativo equilíbrio entre as duas classes S e N apresentando, respectivamente, 84,1% e 88,8%, e uma média ponderada entre as classes de 86,8%.

5. Conclusões e Trabalhos Futuros

O objetivo deste trabalho foi a concepção de um modelo de filtro de *tweets* relevantes para uso em aplicativos voltados a TV Social. Para avaliar o seu desempenho e treinar os modelos de aprendizado de máquina, construiu-se um corpus de *tweets* referentes ao *talent show* culinário MasterChef Brasil.

Embora todas as medidas de avaliação calculadas tenham evidenciado um melhor resultado do classificador SVM frente ao Naive Bayes, para a avaliação final foi utilizada a acurácia, que corresponde ao índice de instâncias classificadas corretamente, ou seja, o percentual de instâncias que o classificador previu corretamente. Com essa medida, obteve-se um resultado de 86,9% com o classificador SVM e 80,6% utilizando o algoritmo Naive Bayes. Outras medidas foram apresentadas com o objetivo de análises pontuais relativas ao algoritmo SVM.

Trabalhos relacionados já demonstraram o potencial da utilização de SVM para a tarefa de classificação de *tweets* de acordo com a sua relevância [Pang et al. 2002, Alves et al. 2014]. Considerando o ambiente da rede social e suas limitações de um *tweet* conter, no máximo, 140 caracteres, os resultados aqui apresentados mostram seu potencial para uso em diversas aplicações de TV social.

Além do método de classificação de *tweets* por relevância em textos em português do Brasil e a criação do modelo para uso em aplicações voltadas a TV Social, uma importante contribuição deste trabalho é disponibilizar um corpus que poderá ser utilizado em outras pesquisas de classificação de *tweets*.

Como trabalhos futuros, está em andamento a coleta e normalização de novos *tweets*, em especial os postados no episódio final da temporada, em 22 de agosto de 2017. Esses *tweets*, produzidos no último episódio da temporada, contam com a participação de 250 influenciadores digitais representando, no contexto desse trabalho, os anotadores especialistas na tarefa em questão.

Em termos de metodologia, também está em investigação o uso de *word embeddings* gerados para o português do Brasil e disponibilizados recentemente pelo NILC⁹, no

⁹<http://www.nilc.icmc.usp.br/>. Acesso em: 21 ago. 2017.

processo de classificação de *tweets*.

Agradecimentos

Esse trabalho está relacionado ao projeto MMeaning, apoiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo nº 2016/13002-0 (Auxílio Regular).

Referências

- Aberdeen, D., Pacovsky, O., e Slater, A. (2010). The learning behind gmail priority inbox. In *NIPS 2010 Workshop on Learning on Cores, Clusters and Clouds*.
- Alves, A. L. F., Baptista, C. S., Firmino, A. A., Oliveira, M. G., e Paiva, A. C. (2014). A Comparison of SVM Versus Naive-Bayes Techniques for Sentiment Analysis in Tweets: A Case Study with the 2013 FIFA Confederations Cup. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web, WebMedia '14*, pages 123–130, New York, NY, USA. ACM.
- Cesar, P. e Geerts, D. (2011). Understanding Social TV: a survey. In *Proceedings of the Networked and Electronic Media Summit (NEM Summit 2011)*.
- Cobo, A., Parra, D., e Navón, J. (2015). Identifying Relevant Messages in a Twitter-based Citizen Channel for Natural Disaster Situations. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion*, pages 1189–1194, New York, NY, USA. ACM.
- Jindal, N. e Liu, B. (2008). Opinion Spam and Analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM '08*, pages 219–230, New York, NY, USA. ACM.
- Klym, N. e Montpetit, M. J. (2008). Innovation at the edge: Social TV and beyond. *MIT Communications Futures Program*.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.
- Pagani, M. e Mirabello, A. (2011). The Influence of Personal and Social-Interactive Engagement in Social TV Web Sites. *International Journal of Electronic Commerce*, 16(2):41–68.
- Pang, B., Lee, L., e Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP 2002*, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Pennacchiotti, M. e Popescu, A.-M. (2011). Democrats, Republicans and Starbucks Aficionados: User Classification in Twitter. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, pages 430–438, New York, NY, USA. ACM.
- Souza, M. e Vieira, R. (2012). Sentiment analysis on twitter data for portuguese language. In *International Conference on Computational Processing of the Portuguese Language*, pages 241–247. Springer.

- Tsugawa, S., Kikuchi, Y., Kishino, F., Nakajima, K., Itoh, Y., e Ohsaki, H. (2015). Recognizing Depression from Twitter Activity. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, pages 3187–3196, New York, NY, USA. ACM.
- Wang, H., Zhang, C., Li, M., e Ji, Y. (2014). Social TV real-time chatting application design. In *Proceedings of the International Symposium on Wireless Personal Multimedia Communications (WPMC 2014)*, pages 209–214.

Detecção de mácula em imagens de fundo de olho por meio da metaheurística *Firefly*

José O. Lins¹, Rodrigo Veras¹, Leonardo Sousa¹

¹Departamento de Computação – Universidade Federal do Piauí (UFPI)

jose.lins.neto182@gmail.com, {rveras,leonardosousa}@ufpi.edu.br

Abstract. *One of the clinical examinations for detecting ocular diseases is the fundus image analysis in search of artifacts. The accumulation of these on the macular region determine the existence of bad clinical picture. In this paper we propose a system for automatic detection of macula by both, image processing techniques and the metaheuristic firefly, a evolutionary algorithm that aims to global maximum with its fitness function. Parameters adjustments were made in order to propose a adaptive method for different scenarios. The tests were made with 94,89% success rate and three public image bases, with a total of 665 images.*

Resumo. *Um dos exames clínicos para detecção de doenças oculares consiste na análise de uma imagem do olho em busca de artefatos. O acúmulo destes nas regiões da mácula determinam a existência de um quadro clínico grave. Neste trabalho é proposto um método de localização automática da mácula por aplicação conjunta de técnicas de processamento de imagens e da metaheurística Firefly, um algoritmo evolutivo que visa atingir o máximo global por meio de sua função de aptidão. Os parâmetros foram ajustados a fim de propor um método que seja adaptável aos diferentes cenários. Os testes foram realizados com uma taxa de sucesso de 94,89% em três bases de imagens públicas, totalizando 665 imagens.*

1. Introdução

No campo oftalmológico, em particular o exame de fundo de olho, muitas características patológicas são pouco visíveis a olho nu. Por este motivo, a localização de artefatos que caracterizem doenças exige muita experiência e concentração dos médicos. Dessa forma, diagnósticos podem divergir entre profissionais, os quais podem ser influenciados pelo grau de fadiga. Programas de computadores podem ajudar no exame de fundo de olho, facilitando a detecção de características patológicas. Isto torna o procedimento menos exaustivo e menos dependente da experiência do médico. Além disso, podem ser usados como ferramenta de triagem.

Das vantagens que emergiram a partir da colaboração entre sistemas computacionais e a medicina, surgiram os sistemas CAD (*Computer Aided Diagnosis*). Sistemas que têm por objetivo minimizar o ônus de trabalhos repetitivos, anteriormente atribuídos como competência dos médicos. A análise de imagens de fundo de olho é um dos métodos de detecção para doenças oculares [Tang et al. 2013]. Utilizando o conhecimento de um

especialista aliado às técnicas de processamento de imagens é possível criar um sistema CAD que auxilie no exame de fundo de olho.

O olho possui duas regiões de suma importância para seu funcionamento, a mácula e o disco óptico (DO). Com notável importância para a visão, a mácula é responsável pela visão central, na qual a imagem é mais nítida. Além disso, no centro da mácula está a fóvea, região com grande concentração de cones, células fotorreceptoras responsáveis pela visão colorida. As células fotorreceptoras têm por função captar um estímulo luminoso e, a partir dele, gerar um estímulo nervoso. Este será direcionado ao cérebro, onde a informação da imagem será processada. Por esse motivo, uma doença que atinge a mácula pode impedir ou dificultar a captação do estímulo luminoso. Desse modo, a informação da imagem pode chegar de forma incompleta ou até mesmo não chegar ao cérebro, causando cegueira parcial ou total.

Pode-se concluir que a localização correta da mácula em algoritmos de diagnóstico de doenças oculares é imprescindível para a determinação da gravidade de uma doença. Neste trabalho, obteve-se uma taxa de sucesso de 94,89%, segundo a métrica de avaliação descrito na Subseção 3.2. Dentre os sistemas CAD existem aqueles cujo objetivo é obter um diagnóstico por meio de do exame de fundo de olho, dentre eles obtiveram melhor resultado aqueles que seguem os passos do diagrama da Figura 1.

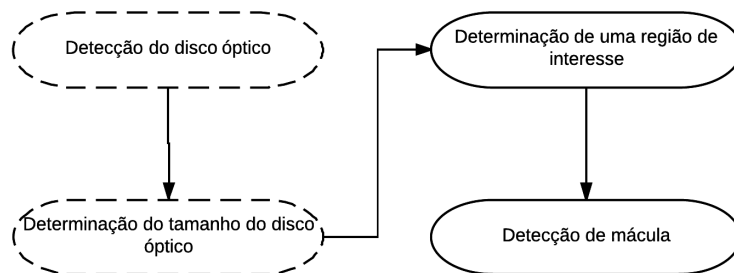


Figura 1. Visão global do funcionamento de um sistema CAD para exame de fundo de olho.

Neste trabalho, as etapas destacadas por elipses de linhas tracejadas ainda não foram automatizadas. Para as duas primeiras etapas do diagrama da Figura 1, utilizou-se dados reais (*ground truth*) fornecidos pela base de imagens para centro e tamanho do DO, quando não fornecidos pela base estes dados foram providos por um médico especialista.

O artigo está dividido como segue: na Seção 2 será feita a revisão bibliográfica de trabalhos relacionados, na Seção 3 serão apresentadas as bases de imagens usadas e a métrica de avaliação do desempenho, na Seção 4 será apresentado o método proposto, juntamente com o pré-processamento e pós-processamento utilizados, na Seção 5 apresenta os resultados e discussão e, por fim, a Seção 6 traz a conclusão do trabalho.

2. Trabalhos Relacionados

Na literatura, várias técnicas foram utilizados a fim de detectar a mácula, porém muitos trabalhos ainda utilizam um método de força bruta para este fim, ou seja, por meio da análise de todo o espaço de busca com utilização de filtros espaciais. Este artigo

avalia o uso do algoritmo *Firefly* como metaheurística para detecção de mácula. Para isso, compararam-se resultados obtidos por estratégia força bruta com o algoritmo de otimização (*Firefly*). Objetiva-se diminuir o custo computacional em processamento de imagens, sem prejudicar a taxa de sucesso obtida.

Os algoritmos força bruta selecionados para comparação com os resultados aqui apresentados são os mesmos de [Veras et al. 2014], os quais foram selecionados por utilizarem a definição de uma região de interesse (RoI - *Region Of Interest*), parte da imagem de possível localização da mácula. Os melhores desempenhos em detectar a mácula foram obtidos utilizando a definição da RoI como um passo anterior à busca pelo centro da mácula [Schwartz and Loewenstein 2015].

Os algoritmos descritos a seguir usam como informação para detecção da mácula a baixa intensidade dos *pixels* presentes nela. Dentre as técnicas utilizadas, [Liu et al. 1997] optaram pela busca da região mais escura e que tenha tamanho de, aproximadamente, 0,5 vezes o tamanho do diâmetro do DO (0,5 DD). De forma semelhante, em [Tan et al. 2009], selecionam-se os *pixels* cuja intensidade seja a mesma e que estejam longe o suficiente para não serem considerados um *cluster*. Em seguida, utiliza-se uma média de intensidades, dentro de uma janela centralizada em cada um destes *pixels*, para definir qual delas representa a mácula. Os autores em [Sagar et al. 2007] aplicam a detecção de vasos sanguíneos para removê-los da RoI e agrupamento dos 1% *pixels* de menor intensidade, em seguida o maior *cluster* será considerado a mácula.

O método de detecção de mácula de [Sekhar et al. 2011] usa um limiar adaptativo e uma operação morfológica de abertura para identificar a mácula. De modo semelhante, [Raja and Ravichandran 2014] usam um limiar adaptativo para segmentação da fóvea, com a diferença que, antes de incrementar o valor do limiar, fazem um teste de aceitação com base em 3 critérios: área, densidade e distância. Caso a região segmentada seja rejeitada no teste de aceitação, incrementa-se o limiar. Os incrementos param quando a quantidade de *pixels* segmentados chega a um valor estimado para o tamanho da fóvea. Após o fim do limiar adaptativo, testam-se as regiões candidatas com base na quantidade de *pixels* referentes a vasos sanguíneos.

Além da busca por *pixels* de baixa intensidade, alguns autores utilizaram também a informação de que a mácula é uma região com baixa incidência de vasos sanguíneos, entre eles estão o algoritmo de [Veras et al. 2014], em que utiliza-se segmentação de vasos sanguíneos e média de intensidade. Calcula-se esta média somente em janelas cuja quantidade de *pixels* referentes a vasos sanguíneos esteja entre as três menores. Já em [Deka et al. 2015] os autores utilizaram equalização adaptativa com contraste limitado (CLAHE - *Contrast Limited Adaptive Histogram Equalization*), operação morfológica de fechamento, inversão da imagem e crescimento de região baseado em uma semente selecionada pelo usuário.

No algoritmo de [Medhi and Dandapat 2015], aplicou-se a transformada *black top hat*, a inversão de intensidade, o limiar de Otsu's e, por fim, buscou-se uma mudança repentina de contraste entre janelas consecutivas, por meio de uma convolução partindo do centro às extremidades, um sentido por vez. A região fora da mudança repentina de contraste é rotulada com intensidade zero, aplica-se a detecção de bordas e a transformada circular de Hough para determinar a região de mácula.

O algoritmo *Firefly* tem sido usado como passo anterior à segmentação de imagens. [Rahebi and Hardalac 2016] usam este algoritmo para detecção do DO em imagens de fundo de olho, como pré-processamento utilizaram o filtro da mediana. Além disso, propuseram uma melhoria na equação de movimento de um *Firefly*, para que este sofra menor influência de máximos locais. Obtiveram uma taxa de acerto de 100%, 95% e 94,38% para as bases DRIVE, STARE e DiaRetDB1, respectivamente.

Para automatização do exame de mamografia, [CRUZ et al. 2016] utilizaram o algoritmo *Firefly* para detecção de microcalcificações em imagens mamográficas, para isso utilizaram como pré-processamento a morfologia matemática de dilatação. O estudo foi feito com base no ajuste de parâmetros e obteve 91% e 95% de acerto para imagens densas e não-densas, respectivamente.

3. Materiais e Métodos

3.1. Bases de Imagens

A base de imagens DRIVE (*Digital Retinal Images for Vessel Extraction*) consiste em 40 fotografias de fundo de olho, das quais 5 delas não foi possível, visualmente, detectar a mácula, e por este motivo foram excluídas. As imagens foram capturadas em formato digital por uma câmera não midriática Canon CR5 3CCD com um campo de vista de 45°. Estas imagens possuem dimensões de 768 x 584 *pixels*, utilizam oito bits por canal de cor e estão no formato TIFF.

A base de imagens ARIA (*Automated Retinal Image Analysis*) possui um total de 116 imagens, sendo 61 de retinas saudáveis e 55 de retinas com alguma patologia. As imagens da base ARIA foram obtidas de uma câmera Zeiss FF450+ sob o formato JPG e utilizam 8 bits por canal de cor. A base ARIA possui ainda uma vasta diversidade de patologias nas imagens e grandes variações de luminosidade, o que aumenta a dificuldade no desenvolvimento de algoritmos robustos para detecção do DO e da mácula.

A base de imagens MESSIDOR (*Methods to evaluate segmentation and indexing techniques in the field of retinal ophthalmology*) foi criada pelo Ministério Francês da Pesquisa e Defesa com o objetivo de facilitar estudos para diagnóstico de retinopatia diabética auxiliado por computador. Esta base é composta por um total de 1200 imagens capturadas com a câmera de vídeo TopCon TRC-NWC com uma angulação de 45° com relação ao campo de visão e possuem 8 bits por canal de cor, sendo que 800 imagens apresentam dilatação de pupila e outras 400 não.

As imagens estão divididas em três conjuntos de dados, um conjunto para cada hospital de oftalmologia, e possuem dimensões de 1440 x 960, 2240 x 1488 *pixels*. Para cada imagem, os médicos especialistas forneceram os diagnósticos para a retinopatia diabética e para o risco de edema macular. Porém, não há informações da quantidade e localização das lesões. No total a base possui 660 imagens de retinas saudáveis e 540 de retinas com patologia. Neste artigo, somente 514 imagens da base MESSIDOR foram utilizadas nos testes dos algoritmos de detecção de mácula, por serem as únicas que contém dados suficientes para realização dos testes.

3.2. Métrica de Avaliação de Desempenho

Uma metodologia para avaliar a performance dos métodos de detecção da mácula foi apresentada por [Veras et al. 2014], na qual definiram três *status* para os resultados: sucesso,

satisfatório e falha. Para definir a qual dos *status* pertence uma saída de um algoritmo, comparam-se as coordenadas obtidas computacionalmente (x_E, y_E) com o par de coordenadas real do centro da mácula (x_M, y_M) .

Define-se sucesso quando a distância euclidiana entre as coordenadas (x_E, y_E) e (x_M, y_M) for menor ou igual a 0,25 DD, como definido na Equação 1. Um resultado satisfatório ocorre quando a distância euclidiana está entre 0,25 DD e 1 DD, segundo a Equação 2. Para valores de distância que superam 1 DD, a saída do algoritmo é rotulada como falha. A Figura 2 apresenta os possíveis resultados.

$$\sqrt{(x_M - x_E)^2 + (y_M - y_E)^2} \leq 0,25DD \quad (1)$$

$$0,25DD < \sqrt{(x_M - x_E)^2 + (y_M - y_E)^2} \leq 1DD \quad (2)$$

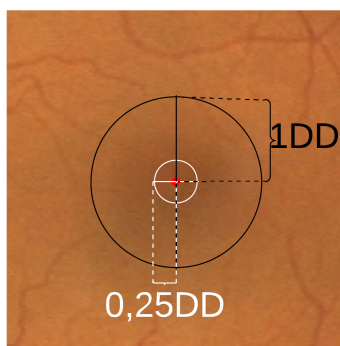


Figura 2. A circunferência branca demarca o limite de 0,25DD, onde o *status* é definido como sucesso, entre a circunferência branca e a preta estão resultados satisfatórios, fora desses limites temos uma falha.

4. Método Proposto

O algoritmo proposto pode ser descritos, de forma geral, em 4 etapas: localização do centro do DO, determinação do tamanho do DO, definição da RoI e localização do centro da mácula. A Figura 3 apresenta todas as subetapas do algoritmo proposto a partir da determinação do centro do DO. Suprimiu-se as duas primeiras etapas na Figura 3 porque os dados do DO foram obtidos do *ground truth*, como dados de entrada.

4.1. Definição da RoI

Para corte da região de interesse (RoI), utilizou-se o método descrito em [Veras et al. 2014], no qual traça-se um segmento de reta entre o centro do DO e o centro da imagem, a partir do ponto médio deste segmento, desloca-se verticalmente 1DD para cima e define-se o ponto esquerdo superior de um quadrado de lado 2DD, rotula-se este quadrado como RoI. A Figura 4 apresenta o método de corte citado.

4.2. Segmentação dos Vasos

Pixels referentes a vasos sanguíneos e à mácula geralmente apresentam intensidades próximas. A fim de diminuir as chances de um *Firefly* ficar preso em um *pixel* referente

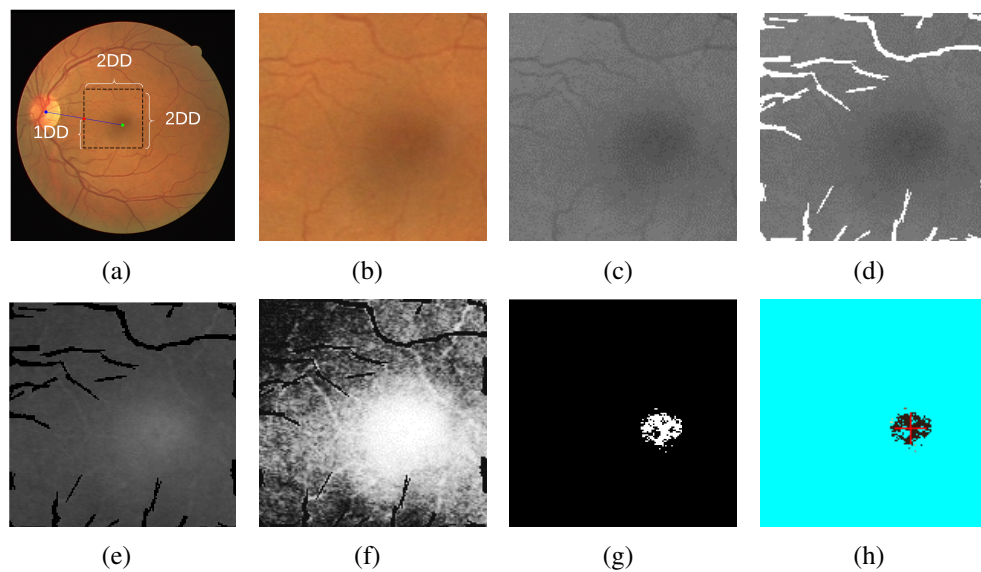


Figura 3. O passo a passo da metodologia pode ser descrita em 8 etapas principais, representadas nas figuras que seguem, 3(a) definição da RoI, 3(b) recorte de parte da imagem, referente à RoI, 3(c) separação da banda de cor G do modelo RGB, 3(d) segmentação de vasos sanguíneos, 3(e) inversão de intensidades, 3(f) equalização de histograma, 3(g) segmentação da mácula e 3(h) definição do centro da mácula.

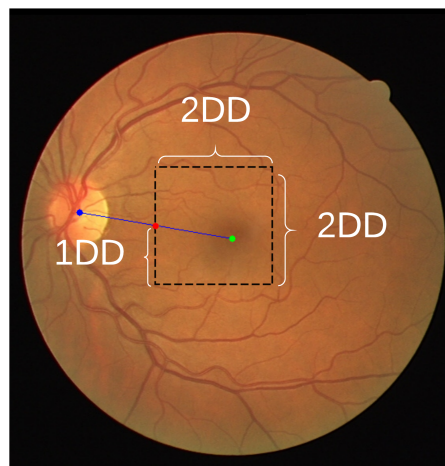


Figura 4. Região de interesse determinada segundo o método de (Veras et al., 2014), o ponto azul marca o centro do DO, o verde o centro da imagem e, por fim, o vermelho marca o ponto médio do segmento.

a um vaso sanguíneo do olho, que seja máximo local, ou até mesmo global, utilizou-se o método descrito em [Zana and Klein 2001] para segmentação de *pixels* referentes a vasos sanguíneos.

4.3. Suavização

Na imagem de fundo de olho, a mácula aparece como a região mais escura, por esse motivo, nas imagens e bandas de cores selecionadas, seus *pixels* apresentam-se com menores intensidades. Dado que o algoritmo proposto busca picos em sua função objetivo, utilizou-se a imagem com intensidades complementares àquela usada como entrada.

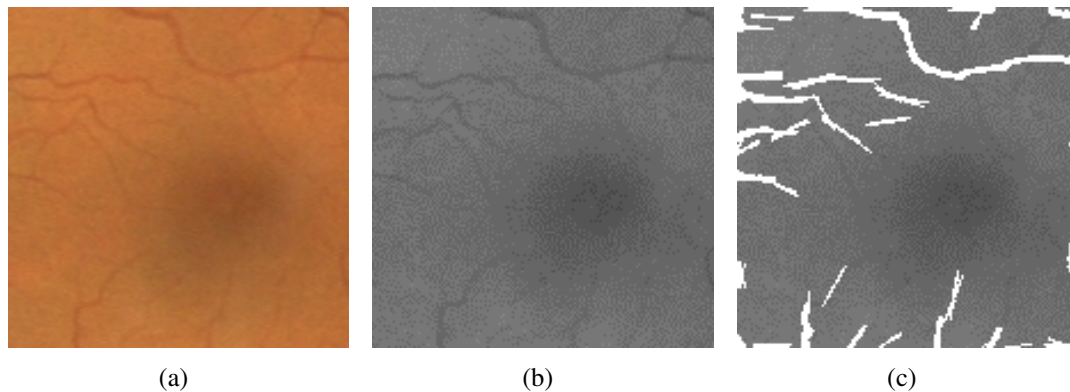


Figura 5. Os resultados obtidos após corte da RoI, separação da banda G do modelo RGB e segmentação de vasos estão representados pelas Figuras 5(a), 5(b) e 5(c), respectivamente.

Desse modo, tanto a mácula passa a ter *pixels* com maiores intensidades, como os *pixels* pertencentes a vasos sanguíneos, que foram segmentados no passo anterior, passam a ter menores intensidades.

Frequentemente, imagens de fundo de olho têm problemas de diferença de iluminação e interferência da luz externa, ocasionando o surgimento de ruídos. Ruídos podem prejudicar o desempenho de algoritmos de detecção de mácula. Um *pixel* ruidoso, sendo ele um pico, pode gerar uma saída errônea para o algoritmo proposto, pois esta saída pode basear-se em uma intensidade que não deveria existir.

Visto a importância da remoção de ruídos, dois tipos de pré-processamento foram testados, são eles: filtro espacial da mediana com janela de tamanho 15 x 15 e equalização de histograma. Ambos causam suavização na imagem. O filtro da mediana gera uma imagem com menores desníveis de intensidade. Essa característica pode vir a prejudicar o desempenho do algoritmo *Firefly*, pois a aproximação de um *Firefly* em relação a um pico depende intrinsecamente da diferença de intensidades entre *fireflies*. Por outro lado, a equalização de histograma mantém desníveis nas intensidades, preservando os picos de intensidade e favorecendo o algoritmo proposto.

A função de aptidão da imagem da RoI para a suavização por filtro da mediana está representada na Figura 6. As coordenadas x e y são as mesmas para os *pixels* da imagem, enquanto o eixo z da Figura 6(b) representa as intensidades de cada *pixel*. A Figura 6(a) é uma visão da projeção da Figura 6(b) no plano xy . Por fim, a Figura 6(c) é o estado final do algoritmo *Firefly*.

4.4. Algoritmo *Firefly*

O algoritmo *Firefly* é baseado no comportamento de vagalumes em período de acasalamento, em que emite-se luz para atrair um parceiro do sexo oposto. Na versão algorítmica deste comportamento, os *fireflies* não possuem sexo e são atraídos pela intensidade da luz. Proposto por [Yang 2010], define-se como algoritmo estocástico de otimização por enxame de partículas (PSO) quando γ tende a zero. Dessa forma, possui uma função de aptidão, um espaço de busca, uma equação de movimento, uma velocidade e um critério de parada.

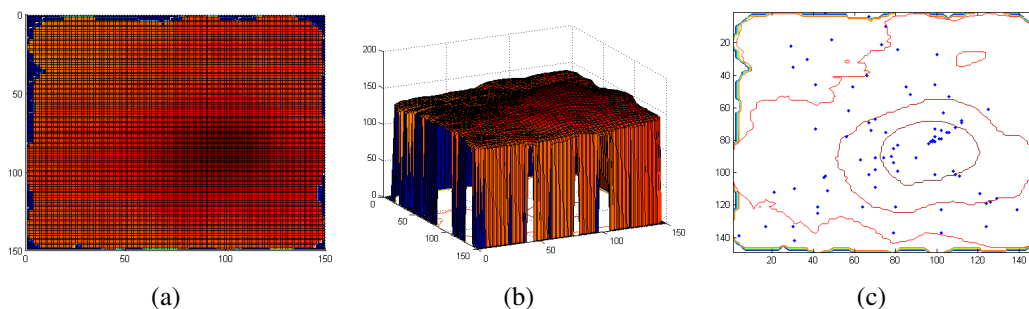


Figura 6. Na Figura 6(a) temos a representação da função de aptidão em 2 dimensões, 6(b) mostra a mesma função em 3 dimensões e 6(c) o estado final do algoritmo para a mesma função, onde os pontos azuis são representações gráficas de posições dos *fireflies* para todas as gerações e sua convergência para o máximo global.

Para processamento digital de imagem, pode-se entender a função objetivo como a própria imagem, em que cada *pixel* é uma das soluções do espaço de busca e sua intensidade é a própria aptidão. A equação de movimento está descrita pelas Equações 3, 4, 5 e 6. Por fim, velocidade e critério de parada são definidos a partir dos parâmetros. Nas Equações 3, 4, 5, 6 e 7, temos o cálculo de distância euclidiana entre dois *fireflies*, do coeficiente de absorção, as atualizações das coordenadas x e y e a diminuição da aleatoriedade, respectivamente.

$$d = \sqrt{(xn_i - xo_j)^2 + (yn_i - yo_j)^2} \quad (3)$$

$$c = \beta \times e^{(-\gamma \times d^2)} \quad (4)$$

$$xn_i = xn_i + c \times (xo_j - xn_i) + \alpha \times (rand - 0.5) \quad (5)$$

$$yn_i = yn_i + c \times (yo_j - yn_i) + \alpha \times (rand - 0.5) \quad (6)$$

$$\alpha = \alpha \times \Delta \quad (7)$$

Em razão de a Equação 4 gerar valores extremamente baixos, necessitou-se alterá-la para a Equação 8, pois a movimentação dos *fireflies* não estava sendo realizada. Isso porque as coordenadas da imagem são valores inteiros e os incrementos ou decrementos obtidos pela Equação 4 estavam em um valor decimal, aproximando-se o valor gerado pela Equação 4 de coordenada para o valor inteiro nenhuma mudança era percebida.

$$c = \beta \times e^{(-\gamma \times d)} \quad (8)$$

Os parâmetros do algoritmo são: quantidade de gerações, quantidade de *fireflies*, fator de aleatoriedade(α), coeficiente de absorção(γ), coeficiente de atração quando $d = 0$ (β) e redução de aleatoriedade(Δ). A quantidade de gerações é usada para definir um critério de parada. O fator de aleatoriedade estimula a maior exploração do espaço de busca. O coeficiente de absorção influencia diretamente na velocidade de deslocamento do *Firefly*. Por último, o α reduz a aleatoriedade no decorrer das gerações (segundo a Equação 7), tornando o deslocamento em direção ao objetivo menos variável. Os valores

testados para cada atributo está definido na Tabela 1. A faixa de valores praticável para γ é entre 0,1 e 10, como descrito em [Fister et al. 2013], entretanto os melhores resultados foram obtidos com os parâmetros: $gerações = 15$, $fireflies = 25$, $\beta = 0,9$, $\alpha = 0,52$, $\gamma = 0,05$, $\Delta = 0,77$.

Tabela 1. Parâmetros do algoritmo *Firefly*.

Parâmetros avaliados	Valor
Gerações	15
<i>Fireflies</i>	25
β	0,9
α	0,32 a 0,52
γ	0,05, 0,06 e 0,1 a 10
Δ	0,77

Uma única imagem da base MESSIDOR pode ter uma resolução de 2240 x 1488, o raio do DO em uma imagem desta base chega a medir 104 *pixels*, portanto define-se uma ROI com resolução 417 x 417, que, por sua vez, gera um espaço de busca de 173.889 *pixels*. Apesar do tamanho desse espaço de busca, com o algoritmo *Firefly* serão visitados apenas $25 \times 25 \times 15 = 9375$ *pixels*, devido aos parâmetros de 25 *fireflies* e 15 gerações. Isso representa um grande ganho de desempenho em relação a algoritmos que aplicam filtros espaciais, que através de convoluções visitam cada *pixel* da imagem, uma ou mais vezes, dependendo da sobreposição entre duas janelas consecutivas. O algoritmo *Firefly* está descrito de forma simplória em Algoritmo 1.

Algoritmo 1: Algoritmo *Firefly*.

Entrada: Imagem, qtGeracoes, qtFireflies, α , γ , Δ

Saída: Vetor de intensidades

```

1 início
2   Inicialização da posição dos fireflies;
3   enquanto  $t < qtGeracoes$  faça
4     Ordenação dos fireflies;
5     para cada Firefly faça
6       if  $Intensidade_i < Intensidade_j$  then
7         | Movimentar Firefly;
8       end
9     fim
10    Diminuir aleatoriedade;
11  fim
12 fim
```

4.5. Limiarização

A saída do algoritmo *Firefly* é um vetor de intensidades, cujos valores foram obtidos dos *pixels* referentes a cada *firefly*. Após inversão da imagem, os *pixels* de interesse para localização da mácula tornam-se aqueles que possuem maiores intensidades, por esse motivo utilizou-se o maior valor de intensidade dentre os *fireflies* e segmentou-se os *pixels* que continham intensidade igual ou maior a este valor. Uma vez segmentada a imagem,

é possível que mais de um aglomerado de *pixels* esteja presente. A fim de determinar qual destes aglomerados de *pixels* corresponde à mácula, selecionou-se aquele que possui a maior área. Finalmente, definiu-se o centroide da componente conectada selecionada. Por meio do método de avaliação de desempenho descrito na Subseção 3.2, definiu-se um *status* para a coordenada obtida pelo algoritmo.

5. Resultados e Discussão

Uma análise prévia sobre qual banda de cor apresenta melhor desempenho na localização da mácula foi feita em [Lins et al. 2016], no qual foram avaliadas as bandas R e G, do modelo de cores RGB, a imagem em níveis de cinza e a fusão das bandas R e G, do modelo RGB. Além disso, também foram testadas as bandas: S e V do modelo HSV, V do modelo Luv e Y do modelo YCbCr.

Os melhores resultados foram obtidos na banda G do modelo RGB, com exceção da base ARIA, que teve resultados sutilmente melhores para a banda Y do modelo YCbCr. A Tabela 2 apresenta os resultados obtidos para o algoritmo *Firefly* na banda de cor G nas diferentes bases, em que utilizou-se filtro da mediana como técnica de remoção de ruídos. Por outro lado, a Tabela 3 apresenta os resultados obtidos por equalização de histograma. As letras S e D que aparecem após os nomes das bases referem-se a conjuntos de imagens definidas como saudáveis ou doentes, respectivamente.

Tabela 2. Resultados do algoritmo proposto usando filtro da mediana como pré-processamento em (%).

	Sucesso	Satisfatório	Falha
DRIVE	97,14	0,00	2,86
ARIA S	88,52	6,56	4,92
ARIA D	67,27	7,27	25,46
MESSIDOR S	99,20	0,40	0,40
MESSIDOR D	97,74	0,00	2,26
Todas	94,59	1,65	3,76

Tabela 3. Resultados do algoritmo proposto usando equalização de histograma como pré-processamento em (%).

	Sucesso	Satisfatório	Falha
DRIVE	97,14	0,00	2,86
ARIA S	85,24	8,20	6,56
ARIA D	60,00	7,27	32,73
MESSIDOR S	100,00	0,00	0,00
MESSIDOR D	98,12	0,38	1,50
Todas	94,89	1,80	3,31

Os resultados obtidos para a base ARIA tiveram taxa de sucesso superior com utilização do filtro da mediana como pré-processamento, porém a base MESSIDOR é mais expressiva, por conter maior quantidade de imagens. A última linha das Tabelas 2 e 3 representa a união das 665 imagens, composta pelas 3 bases: DRIVE, ARIA e MESSIDOR. Usando essas 665 imagens, avaliou-se as taxas dos 3 possíveis *status* usando o algoritmo proposto. Posteriormente, comparou-se os melhores resultados obtidos com aqueles citados em [Veras et al. 2014], no qual foram usados no total 254 imagens, em que existiam somente 103 imagens da base MESSIDOR. A Tabela 4 apresenta esses valores.

Tabela 4. Comparação de desempenho do algoritmo proposto em relação a outros algoritmos da literatura em (%).

	Sucesso	Satisfatório	Falha
Liu et al.	62,63	5,05	32,32
Sagar et al.	61,62	13,13	25,26
Tan et al.	80,81	11,11	8,08
Sekhar et al.	30,30	50,51	19,19
Veras et al.	81,12	15,67	3,22
Método proposto	94,89	1,80	3,31

Apesar da desvantagem do filtro da mediana descrita na Seção 4.3, este filtro obteve os melhores resultados em uma das bases e valor muito próximo ou igual ao melhor resultado nas outras duas. Isso ocorre porque uma vez que exista diferença de intensidades entre dois *fireflies* a movimentação ocorre. O filtro da mediana pode causar uma diminuição na taxa de sucesso caso ocorra uma expansão das baixas intensidades de *pixels* da mácula para fora desta região, como consequência o centroide da componente conectada selecionada após a limiarização pode sofrer um deslocamento, distanciando-se do centro da mácula. O método proposto obteve taxa de sucesso maior do que os outros métodos apresentados em [Veras et al. 2014]. Atribui-se este resultado tanto à eficiência do algoritmo, quanto ao uso de dados reais de centro do DO e seu tamanho.

6. Conclusão

O algoritmo proposto apresenta ganho significativo em termos de custo computacional, se comparado a estratégias de detecção de mácula que não utilizam métodos heurísticos, pois à cada geração do algoritmo somente os 25 *fireflies* serão comparados. Executou-se o programa em máquina com processador AMD A10-5800B APU, placa de vídeo AMD Radeon HD 7660D e 8GB de RAM DDR3, com essa configuração a duração para execução do método proposto e pós-processamento foi de 49,746 s para as 665 imagens. Além disso, obteve-se taxa de sucesso superior aos métodos citados em [Veras et al. 2014].

Avaliando-se os resultados obtidos para imagens saudáveis e doentes, percebe-se que a base ARIA tem imagens doentes com maior quantidade de artefatos que prejudicam a detecção da mácula. Por outro lado, na base MESSIDOR a diferença entre resultado para imagens saudáveis e doentes foi significativamente menor. Um menor desempenho é esperado quando avalia-se imagens doentes, porém ajustes precisam ser feitos para que a detecção na base ARIA seja menos sensível a estes artefatos.

Em trabalhos futuros pretende-se automatizar as etapas de detecção do centro e do tamanho do DO e realizar mais testes de ajuste de parâmetros do *Firefly*. Avalia-se também a possibilidade de alterações na função de movimento, semelhante ao proposto por [Rahebi and Hardalaç 2016].

Referências

- CRUZ, Y., OTILIO, P., SILVE, A., and PAIVA, A. (2016). Segmentação de microcalcificações em imagens mamográficas utilizando algoritmo firefly. *SBIS*.
- Deka, D., Medhi, J. P., and Nirmala, S. (2015). Detection of macula and fovea for disease analysis in color fundus images. In *Recent Trends in Information Systems (ReTIS), 2015 IEEE 2nd International Conference on*, pages 231–236. IEEE.

- Fister, I., Yang, X.-S., and Brest, J. (2013). A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation*, 13:34–46.
- Lins, J., Veras, R., Vogado, L., Aires, K., Machado, V., and Leocadio, J. (2016). Análise do uso de diferentes bandas de cores na detecção da mácula em imagens de retina. *CBA*.
- Liu, Z., Opas, C., and Krishnan, S. M. (1997). Automatic image analysis of fundus photograph. In *Engineering in Medicine and Biology Society, 1997. Proceedings of the 19th Annual International Conference of the IEEE*, volume 2, pages 524–525. IEEE.
- Medhi, J. P. and Dandapat, S. (2015). Automatic detection of fovea using property of vessel free region. In *Communications (NCC), 2015 Twenty First National Conference on*, pages 1–6. IEEE.
- Rahebi, J. and Hardalaç, F. (2016). A new approach to optic disc detection in human retinal images using the firefly algorithm. *Medical & biological engineering & computing*, 54(2-3):453–461.
- Raja, J. B. and Ravichandran, C. (2014). Automatic localization of fovea in retinal images based on mathematical morphology and anatomic structures. *International Journal of Engineering and Technology*, 6(5):2171–2183.
- Sagar, A. V., Balasubramanian, S., and Chandrasekaran, V. (2007). Automatic detection of anatomical structures in digital fundus retinal images. In *Machine Vision Applications*, pages 483–486.
- Schwartz, R. and Loewenstein, A. (2015). Early detection of age related macular degeneration: current status. *International Journal of Retina and Vitreous*, 1(1):1–8.
- Sekhar, S., Abd El-Samie, F. E., Yu, P., Al-Nuaimy, W., and Nandi, A. K. (2011). Automated localization of retinal features. *Applied optics*, 50(19):3064–3075.
- Tan, N., Wong, D., Liu, J., Ng, W., Zhang, Z., Lim, J., Tan, Z., Tang, Y., Li, H., Lu, S., et al. (2009). Automatic detection of the macula in the retinal fundus image by detecting regions with low pixel intensity. In *International Conference on Biomedical and Pharmaceutical Engineering, 2009. ICBPE'09.*, pages 1–5. IEEE.
- Tang, H. L., Goh, J., Peto, T., Ling, B. W.-K., Hu, Y., Wang, S., Saleh, G. M., et al. (2013). The reading of components of diabetic retinopathy: an evolutionary approach for filtering normal digital fundus imaging in screening and population based studies. *PloS one*, 8(7):1–12.
- Veras, R., Silva, R., Aires, K., and Medeiros, F. (2014). Automatic detection of fovea in retinal images using fusion of color bands. In *Conference on Graphics, Patterns and Images (SIBGRAPI), 2014 27th SIBGRAPI*, pages 119–124. IEEE.
- Yang, X.-S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2):78–84.
- Zana, F. and Klein, J.-C. (2001). Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE transactions on image processing*, 10(7):1010–1019.

Estudo Sobre o Impacto do Particionamento dos Dados na Seleção Dinâmica de Atributos

Jhoseph K. L. Jesus^{1,2}, Anne M. P. Canuto¹, Daniel S. A. Araújo²

¹Departamento de Informática e Matemática Aplicada – DIMAp
Universidade Federal do Rio Grande do Norte
Caixa Postal 1524 – CEP 59078-970 – Natal\RN – Brasil

²Instituto Metr pole Digital – IMD
Universidade Federal do Rio Grande do Norte
Caixa Postal 1524 – CEP 59078-970 – Natal\RN – Brasil

jhoseph@ppgsc.ufrn.br, anne@dimap.ufrn.br, daniel@imd.ufrn.br

Abstract. *Recently, data coming from devices and applications that constantly capture and generate data is growing exponentially. Since these applications represent some real scenario, the data produced tends to have high complexity and contains a great amount of features. To handle the raw data, the step of feature selection is an indispensable step to ensure the quality of the analysis. In this context, this paper proposes an exploratory study on a dynamic selection approach in the context of classification tasks. The main purpose is to verify the impact that the change in the measure of similarity and the proportions of data in training, test and validation sets can bring to the accuracy of the system. Results showed that the proposed method presented competitive results compared to other well-known feature selection/extraction algorithms.*

Resumo. *Recentemente, o dados provenientes de dispositivos e aplica es que constantemente capturam e geram dados est crescendo abruptamente. Por se tratar de aplica es que retratam algum cenrio real, os dados produzidos tendem a ter alta complexidade e contm uma grande quantidade de atributos. Para tratar os dados brutos, a etapa de sele o de atributos  uma tarefa indispensvel para garantir a qualidade da anlise. Nesse sentido, este artigo prope um estudo exploratrio sobre uma abordagem de sele o dinmica de atributos existente na literatura no contexto de classifica o de padres. O objetivo  verificar o impacto que a mudana na medida de similaridade usada e nas propor es de diviso dos dados entre os conjuntos de treinamento, teste e valida o pode trazer na acurcia do sistema. Resultados mostraram que o mtodo de sele o dinmica com informa o mtua teve um desempenho competitivo quando comparado a outros mtodos conhecidos de sele o/extra o de caractersticas.*

1. Introdu o

Estudos recentes mostram que a quantidade de dados no mundo vem crescendo exponencialmente, principalmente, depois do advento de dispositivos mveis, como os *smartphones*, por exemplo, que so capazes de produzir e capturar dados de vrios tipos de mdia. Alm disso, a computa o em nuvem trouxe consigo a possibilidade de armazenamento

remoto dos dados, levando a um acúmulo desenfreado desses dados. Com isso qualquer análise manual desses dados se torna inviável, dado a sua grande quantidade. Nesse sentido, o uso de técnicas automáticas para encontrar padrões ou relações entre os dados, como as de Aprendizado de Máquina, se torna essencial [Jain and Dubes 1988].

Vale salientar que os dados são produzidos com fins específicos e armazenados, muitas vezes, sem o propósito de análise posterior. Por causa disso, faz-se necessário um tratamento prévio (pré-processamento) para que possa ser feita uma análise de qualidade [Jain and Dubes 1988]. Um dos métodos mais conhecidos de pré-processamento é a seleção de atributos, que segundo [Guyon and Elisseeff 2003] são responsáveis por tentar filtrar os dados visando obter o subconjunto de atributos que melhor representa todo o conjunto de dados. Esse tipo de abordagem é importante em dois aspectos: pode reduzir drasticamente a quantidade dos dados, provendo um processamento mais rápido; e elimina dados não relevantes que podem causar distúrbios no processo de extração de informação. Sabendo disso, pesquisadores têm desenvolvido, ao longo dos anos, algoritmos para realizar a seleção de atributos em conjuntos de dados [Liu and Motoda 2007].

Paralelo a isso, descritores do campo da Teoria da Informação, inicialmente utilizados para medir a eficiência da transmissão de dados [Shannon 1948], têm sido utilizados com sucesso para quantificar informação em vários problemas reais. Um dos descritores é a Informação Mútua e existem estudos que analisam sua utilização na seleção de atributos para conjuntos de dados complexos. Por exemplo, [Battiti 1994] propôs diversas técnicas baseadas em Informação Mútua para selecionar os atributos mais relevantes de um conjunto de dados. Métodos baseados em Informação Mútua têm vantagem de realizar uma análise de mais alta ordem, não ficando restrita somente a relações lineares entre os dados, tornando-se indicados para situações complexas existentes no mundo real.

Nesse sentido, este artigo propõe um estudo exploratório a abordagem de seleção dinâmica de atributos proposta em [Nunes et al. 2016], que faz uso de técnicas de agrupamento como parte do seu processo, no contexto de tarefas de classificação. Além disso, o método é modificado para incluir a Informação Mútua como medida de similaridade, conforme análises feitas por [Jesus et al. 2017b] e [Jesus et al. 2017a].

2. Método de Seleção de Atributos Avaliado

Como mencionado anteriormente, existem dois métodos principais para realizar o tratamento das características dos dados (*features*): seleção de atributos e extração de atributos. Neste trabalho, vamos ficar restritos a abordagem de seleção de atributos, que busca escolher um subconjunto dos atributos originais que melhor representam todo o espaço de características.

A abordagem de seleção de atributos consiste, inicialmente, na divisão do conjunto de dados em três partes: treinamento, teste e validação. A partir disso, é realizado o agrupamento no conjunto de validação, que cria uma partição $C = c_1, \dots, c_j$ com instâncias similares divididas em grupos. Após isso, para cada grupo, cada atributo tem um valor de relevância associado, considerando a importância (similaridade) desse atributo para o seu grupo.

O trabalho inicial utilizou a correlação de Spearman como medida de similaridade. Neste artigo, utilizamos os estudos anteriores feitos em [Jesus et al. 2017b] e

[Jesus et al. 2017a] e definimos a Informação Mútua para medir a importância dos atributos em relação aos grupos, uma vez que ela apresentou o melhor desempenho dentre todas as medidas avaliadas. O objetivo nesse passo é capturar a quantidade de informação compartilhada por cada atributo em relação ao rótulo da classe.

Uma vez que os atributos mais importantes são selecionados para cada grupo, j classificadores são treinados, um referente a cada grupo. É importante destacar que, nesse momento, utilizamos apenas os atributos selecionados para representar cada grupo. Para testar os modelos, instâncias da partição de teste são providas e, afim de saber quais atributos são válidos para cada instância, as instâncias são associadas, uma a uma, ao grupo mais similar e então classificadas utilizando o modelo definido para aquele grupo. Dessa forma, o processo de seleção de atributos se torna dinâmico, uma vez que as instâncias de teste são avaliadas por diferentes modelos, treinados por diferentes atributos. Todos os detalhes acerca da abordagem podem ser encontrados em [Nunes et al. 2016]. A Figura 1 mostra o funcionamento geral do método.

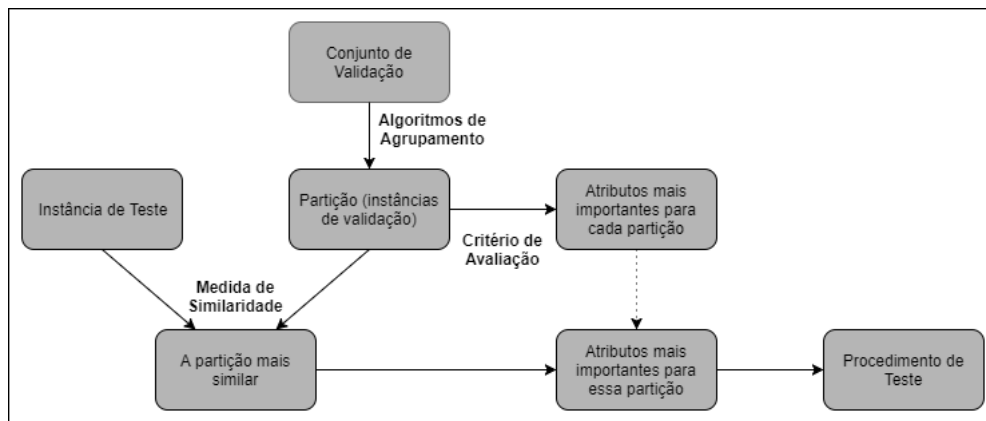


Figura 1. Funcionamento do método de seleção de atributos proposto em [Nunes et al. 2016]

Vale salientar que em todos os trabalhos que utilizaram esse método, a quantidade de instâncias presentes no conjunto de treinamento, validação e teste foram fixas: 80% para treinamento, 10% para teste e 10% para validação. Este trabalho, entre outras coisas, analisa o impacto de mudança dessas proporções.

3. Materiais e Métodos

Esta seção descreve os materiais e métodos utilizados durante a análise dos experimentos conduzidos em relação ao método de seleção dinâmica [Nunes et al. 2016]. As subseções abrangem os seguintes tópicos: Medida Similaridade, que apresenta a definição formal de informação mútua (medida utilizada no método de seleção dinâmica); Algoritmos de Redução de Dimensionalidade utilizados para efeito de comparação em relação a análise realizada; Algoritmos de classificação e Agrupamento utilizados para avaliar o desempenho das abordagens de seleção; Testes Estatísticos executados para validar o desempenho dos métodos de seleção e, por fim, a explanação da configuração do experimento.

3.1. Medida de Similaridade

Uma das tarefas do método de seleção dinâmica de atributos é definir o nível de similaridade entre os atributos da partição de validação e o atributo classe. Diferentemente

do trabalho de [Nunes et al. 2016], onde o método de seleção dinâmica foi proposto utilizando a correlação de Spearman, neste artigo foi utilizado como medida de similaridade um descritor da Teoria da Informação, intitulado de Informação Mútua, uma vez que o uso de tal descritor mostrou um alto impacto positivo no desempenho do método de seleção dinâmica em comparação a utilização da correlação de Spearman [Jesus et al. 2017a, Jesus et al. 2017b]. A Informação Mútua (MI) mensura a quantidade de informação compartilhada entre duas variáveis aleatórias. No contexto de seleção de atributos, essas variáveis representam os atributos a serem selecionados [Shannon 1948]. Em outras palavras, a informação mútua dimensiona a quantidade de informação obtida por um atributo, por meio de outro. Sendo X e Y variáveis aleatórias e p a função de probabilidade. Neste artigo, é considerada a normalidade dos dados. Baseado nisso, a Informação Mútua pode ser definida como:

$$I(X, Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) \quad (1)$$

3.2. Algoritmos de Redução de Dimensionalidade

Para efeito de comparação, foram selecionados dois algoritmos de seleção de atributos baseados em informação mútua: *Conditional Infomax Feature Extraction* (CIFE) [Lin and Tang 2006], que funciona através da maximização da relevância da informação do atributo-classe em detrimento da redução das redundâncias entre os atributos; e *Mutual Information Feature Selection* (MIFS) [Battiti 1994], que é baseado na seleção gulosa de atributos, considerando a informação mútua entre o atributo classe e os atributos selecionados.

Todos os algoritmos de redução de dimensionalidade, bem como o método de seleção dinâmica avaliado, realizaram uma redução de 90% dos atributos para cada base de dados. Em outras palavras, 10% de seleção, para os métodos de seletores (Seleção Dinâmica, CIFE, MIFS e Randômico) e 10% de extração, para os métodos extratores (PCA e LDA). Nos métodos extratores são selecionados 10% dos componentes (para o PCA) e discriminantes (para o LDA) mais representativos.

Adicionalmente aos algoritmos de seleção de atributos baseados em informação mútua, foram utilizados dois algoritmos de extração amplamente utilizados pela comunidade de Aprendizado de Máquina, PCA (não-supervisionado) [Jolliffe 1986] e LDA (supervisionado) [McLachlan 2004], visando comparar os seus desempenhos com o método avaliado.

3.3. Algoritmos de Classificação e Agrupamento

Afim de validar o desempenho das abordagens utilizadas na análise empírica, foram escolhidos dois métodos de classificação: k -nn (método baseado em distância) [Mitchell 1997] e Árvore de Decisão (método baseado em busca) [Mitchell 1997], além de serem amplamente utilizados pela comunidade de Aprendizado de Máquina. Baseado nisso, buscamos utilizar diferentes heurísticas de classificação para evitar resultados tendenciosos para um método específico. Com o intuito de realizar as tarefas de agrupamento no método de seleção dinâmica, foi utilizado o algoritmo de agrupamento k -means, onde o valor de k utilizado corresponde ao valor das classes de cada base de dados. [Kanungo et al. 2002].

3.4. Testes Estatísticos

Afim de comparar a eficácia da abordagem explorada, dois testes estatísticos foram aplicados: Teste de Friedman [Friedman 1937] e o Teste *post-hoc* Wilcoxon Rank-Sum [Gibbons and Chakraborti 2003]. O segundo foi aplicado quando o p-value do teste de Friedman foi menor que 0.05. Ambos os testes foram utilizados com um nível de confiança de 95% ($\alpha = 0.05$).

3.5. Conjuntos de Dados

Para execução dos experimentos, foram selecionadas doze bases de dados de diferentes naturezas. Selecionamos estas bases de dados com o objetivo de cobrir diferentes intervalos de valores em relação ao número de instâncias, classes, atributos e contextos. As bases estão classificadas em diferentes cenários: Processamento de Sinais, Reconhecimento de Padrões (imagens faciais, assinaturas) e bioinformática (análise de expressão gênica).

Os repositórios utilizados para coleta dos dados foram: *UCI Machine Learning* [Lichman 2013], *Arizona State University repository* [Li et al. 2016] e *Bioinformatics Research Group of Seville repository* [Seville 2004]. As principais características de cada conjunto de dados estão apresentadas na Tabela 1, onde n é o número de instâncias, C é o número de classes e a é o número de atributos.

ID	Bases de Dados	n	C	a
B1	LSVT	126	2	310
B2	Lung Cancer	181	2	12533
B3	Breast Cancer Diagnostic	569	2	30
B4	Connectionist Bench	208	2	60
B5	Ionosphere	351	2	32
B6	Lymphoma	96	9	4026
B7	USPS	9298	10	256
B8	PCMAC	1943	2	3289
B9	Friedman	1000	2	100
B10	Colon Cancer	62	2	2000
B11	COIL-20	1440	20	1024
B12	Arrhythmia	452	16	279

Tabela 1. Descrição das Bases de Dados.

3.6. Configuração do Experimento

Os algoritmos utilizados neste artigo foram implementados no *Matlab*. Com o objetivo de obter resultados robustos, na análise empírica foi aplicada a estratégia de validação *10-fold cross-validation* para todos os métodos analisados. Para o método de seleção dinâmica de atributos foram realizadas quatro variações no tamanho das partições de treino e validação, como pode ser observado na Tabela 2. Para os demais métodos, foi utilizado o padrão do *10-fold cross-validation* (nove *folds* de treino e um de *fold* de teste).

Variações	Treino	Validação	Teste
V1	8	1	1
V2	7	2	1
V4	6	3	1
V4	5	4	1

Tabela 2. Variação do tamanho das partições de treino e validação

As bases de dados foram divididas de acordo com cada uma das variantes dos *10-fold cross-validation*. Uma vez que ocorreu essa divisão, foram utilizados o(s) *fold(s)*

de validação para criar os grupos utilizados pelo algoritmo de agrupamento k -means, onde o parâmetro k foi definido igual ao número de classes correspondente a base de dados em execução. Após o processo de agrupamento, os classificadores foram treinados utilizando os respectivos *folds* da partição de treinamento, de acordo com os atributos selecionados pelo método de seleção dinâmica. O *fold* de teste foi usado para avaliar o desempenho do método de seleção dinâmica proposto em [Nunes et al. 2016] e estendido em [Jesus et al. 2017b].

Visando que os métodos analisados são algoritmos não-determinísticos, foram realizadas 10 execuções para cada abordagem, fazendo um total de cem (100) valores de acurácia que foram utilizados como base dos resultados deste artigo. Todas as abordagens analisadas foram executadas utilizando dois algoritmos de classificação, k -NN e Árvore de Decisão, como descrito na seção 3.3, para cada conjunto de dados.

4. Resultados e Discussões

Nesta seção, são apresentados os resultados obtidos com a análise empírica realizada afim de verificar o desempenho dos métodos de seleção dinâmica. A análise empírica foi dividida em três partes, onde a primeira avalia a utilização de quatro variações nos tamanhos das partições de treino e validação para o método de seleção dinâmica (SD-IM). O principal objetivo desta análise é verificar o comportamento do método de seleção dinâmica, quando variando os tamanhos das partições de treino e validação. As duas análises restantes são de caráter comparativo, onde a segunda visa comparar o método SD-IM com outros métodos de seleção de atributos baseados em Informação Mútua e com a seleção de atributos randômica. Por fim, a última análise compara o desempenho do método SD-IM com métodos de extração de atributos conhecidos na literatura, bem como com o desempenho original das bases de dados (sem seleção de atributos). Os resultados estão representados através da acurácia média e desvio padrão obtidos levando em consideração os dois algoritmos de classificação (k -NN e Árvore de Decisão) para cada base dados.

4.1. Análise da Variação no Tamanho das Partições de Treino e Validação

Esta análise tem como principal objetivo definir qual o melhor tamanho, em termos de conjunto de treino e validação, para o método de seleção dinâmica avaliado. Para tanto foram utilizadas quatro variações nos tamanhos das partições que estão representadas na Tabela 3. A nomenclatura (8T1V) implica que foram utilizadas oito partes, *folds*, para treino e uma para validação. A mesma ideia se aplicam as demais (7T2V, 6T3V e 5T4V). Nesta tabela, as células com fundo cinza e em negrito representam o melhor desempenho para cada base de dados (linha).

A partir da Tabela 3, podemos observar que o método de seleção dinâmica com informação mútua (SD-IM) que obteve o melhor desempenho foi o que utilizou a configuração 8T1V, em oito bases de dados. Para as demais bases de dados, a configuração 7T2V alcançou o melhor desempenho em três bases de dados restantes. Além disso, nenhum desempenho superior para 6T3V e somente um para 5T4V.

Afim de obter mais significância para os resultados encontrados nesta análise empírica, foram utilizados dois testes estatísticos, o teste de *Friedman* e o teste post-hoc *Wilcoxon*. Estes testes não-paramétricos são recomendados para comparar o desempenho

Análise da Variação no Tamanho das Partições de Treino e Validação				
Métodos	SD-IM (8T1V)	SD-IM (7T2V)	SD-IM (6T3V)	SD-IM (5T4V)
Métricas	Média±Std	Média±Std	Média±Std	Média±Std
B1	77.38±13.54	71.17±12.26	73.79±12.43	73.46±15.18
B2	96.62±4.73	95.72±4.56	95.39±4.57	95.91±5.01
B3	97.18±6.66	87.58±4.32	88.87±4.27	86.14±5.12
B4	94.27±9.12	67.36±9.00	68.81±11.17	68.33±10.36
B5	98.81±2.78	85.71±6.99	83.77±4.99	86.85±5.92
B6	78.00±18.33	71.90±21.15	75.97±20.59	74.94±20.11
B7	76.49±2.37	80.29±2.50	80.20±2.84	79.60±2.88
B8	83.40±4.27	84.39±5.41	82.80±5.14	82.73±3.92
B9	99.95±0.25	77.73±8.57	77.76±8.64	77.38±6.88
B10	73.39±15.64	73.87±16.94	72.13±18.09	71.70±15.06
B11	93.46±5.93	92.11±7.02	91.31±7.30	91.08±7.64
B12	53.47±7.90	52.25±7.30	54.21±6.96	54.51±6.86

Tabela 3. Resultados Utilizando Diferentes Tamanhos das Partições de Treino e Validação

de diferentes algoritmos de classificação. O teste de *Friedman* foi utilizado para comparar o desempenho de todas as quatro variações no tamanho das partições de treino e validação. Sendo assim, os testes foram aplicados diretamente sobre os níveis de acurácia.

Nestes testes, todas as variações no tamanho das partições foram consideradas com as 10 execuções do *10-fold cross validation*, para os dois algoritmos de classificação. Os resultados do teste de *Friedman* estão apresentados na Tabela 4, onde a primeira coluna representam o *p-value* do teste de *Friedman*. As demais colunas da tabela apresentam os resultados do teste post-hoc, comparando, numa base dois-por-dois, o desempenho do SD-IM (8T1V) com as três variações do SD-IM (7T2V, 6T3V e 5T4V). No teste post-hoc, a comparação A-B foi realizada e ' $<$ ' representa significância estatística superior de A sobre B; ' $>$ ' representa o oposto (significância estatística superior de B sobre A), enquanto '=' representa que não houve evidências para afirmar há diferença entre o desempenho dos métodos A e B.

Na Tabela 4, as células com fundo em cinza e com os números em negrito representam os valores onde a variação do método de seleção dinâmica com Informação Mútua utilizando (8T1V) obteve desempenho superior, do ponto de vista estatístico. Em contraste, os números seguidos por (*) representam os casos onde as demais variações obtiveram desempenho superior ao método SD-IM (8T1V). Para as demais células, o desempenho de ambas as variações são similares, do ponto de vista estatístico.

A partir da Tabela 4, podemos observar que as configurações de variação para o número de partições de treino e validação são diferentes, do ponto de vista estatístico, em 11 bases de dados (*p-value* < 0.05). Em apenas uma base de dados, B10, o desempenho dos diferentes números de partições de treino e validação são similares do ponto de vista estatístico. No teste *post-hoc*, podemos afirmar que a configuração 8T1V obteve desempenho superior (células sombreadas com números em negrito) em relação as demais configurações, do ponto de vista estatístico, em 8 bases de dados, quando comparado com a configuração 7T2V; em 7 bases de dados, quando comparado com a configuração 6T3V; e em 6 bases de dados, quando comparado com a configuração 5T4V. Os resultados obti-

Métodos	Testes Estatísticos - Variação do Tamanho das Partições de Treino e Validação			
	Friedman	(8T1V) - (7T2V)	(8T1V) - (6T3V)	(8T1V) - (5T4V)
Métricas	p-value	p-value	p-value	p-value
B1	1.75E-06	<	<	<
B2	0.00433	<	<	=
B3	1.25E-75	<	<	<
B4	3.78E-82	<	<	<
B5	4.34E-92	<	<	<
B6	2.20E-05	<	=	=
B7	6.85E-38	>(*)	>(*)	>(*)
B8	6.07E-11	>(*)	=	=
B9	1.21E-98	<	<	<
B10	0.43566	-	-	-
B11	3.61E-23	<	<	<
B12	0.002699	=	=	=

Tabela 4. Resultados dos Testes Estatísticos Utilizando Diferentes Tamanhos das Partições de Treino e Validação

dos ratificam o bom desempenho obtido pela configuração 8T1V em termos de acurácia. Sendo assim, a seguir, o método de seleção dinâmica com Informação Mútua utilizando a configuração 8T1V será utilizado nas demais análises feitas.

4.2. Análise de Diferentes Métodos de Seleção de Atributos

O objetivo desta seção é comparar o desempenho do método SD-IM utilizando a configuração 8T1V, com outros métodos de seleção de atributos baseados em informação mútua, bem como com a seleção randômica de atributos. Os algoritmos baseados em informação mútua escolhidos foram: CIFE e MIFS (descritos na seção 3.2). A Tabela 5 descreve os resultados obtidos pelos quatro métodos, baseados na acurácia média dos classificadores. Nesta tabela, os resultados com valor em negrito e cor de fundo sombreada representam o melhor método de seleção (com maior acurácia) para cada base de dados.

Métodos	Análise Comparativa - Métodos de Seleção de Atributos			
	SD- (8T1V)	CIFE	MIFS	RandomFS
Métricas	Média±Std	Média±Std	Média±Std	Média±Std
B1	77.38±13.54	72.39±6.83	68.71±7.37	65.36±11.02
B2	96.62±4.73	95.86±2.05	93.01±2.71	98.10±2.03
B3	97.18±6.66	90.93±3.04	92.95±1.94	83.86±3.38
B4	94.27±9.12	75.70±4.12	71.81±4.20	56.67±4.08
B5	98.81±2.78	87.90±2.81	86.78±2.61	81.13±2.79
B6	78.00±18.33	73.07±6.39	75.70±13.56	78.22±15.59
B7	76.49±2.37	85.50±2.73	90.17±4.59	87.26±3.84
B8	83.40±4.27	78.58±3.01	88.16±1.82	67.60±2.30
B9	99.95±0.25	81.09±6.76	74.03±8.48	57.35±3.29
B10	73.39±15.64	71.90±10.59	72.42±7.93	77.15±9.40
B11	93.46±5.93	94.81±5.09	95.09±4.86	96.08±4.01
B12	53.47±7.90	55.03±5.17	28.77±26.69	56.01±4.49

Tabela 5. Resultados Utilizando Outros Métodos de Seleção de Atributos

Ao analisar a Tabela 5, podemos observar que o método avaliado (SD-IM 8T1V) obteve o melhor desempenho dos métodos de seleção em cinco bases de dados, assim

como o método de seleção randômica, que obteve em cinco bases de dados. Enquanto que o método MIFS alcançou desempenho superior em duas bases de dados e o CIFE em nenhuma. Desta forma, podemos perceber que o SD-IM obteve desempenho notadamente superior que o MIFS e CIFE, além de desempenho similar ao método randômico, analisando apenas as acurácias obtidas.

Na Tabela 6 estão apresentados os resultados dos testes estatísticos, comparando os quatro métodos de seleção de atributos sob as perspectivas dos testes de *Friedman* (primeira coluna) e o teste *post-hoc* (demais colunas, comparando o método avaliado [SD-IM 8T1V] com os demais métodos de seleção de atributos), para todas as bases de dados.

Testes Estatísticos - Métodos de Seleção de Atributos				
Métodos	Friedman	(SD-IM 8T1V) - (CIFE)	(SD-IM 8T1V) - (MIFS)	(SD-IM 8T1V) - (RFS)
Métricas	p-value	p-value	p-value	p-value
B1	3.91E-37	<	<	<
B2	4.34E-46	<	<	=
B3	1.37E-121	<	<	<
B4	6.23E-141	<	<	<
B5	8.22E-136	<	<	<
B6	1.01E-08	<	<	=
B7	2.18E-155	>(*)	>(*)	>(*)
B8	7.86E-154	<	>(*)	<
B9	1.31E-163	<	<	<
B10	3.18E-09	=	=	>(*)
B11	1.31E-71	>(*)	>(*)	>(*)
B12	1.82E-65	>(*)	<	>(*)

Tabela 6. Resultados dos Testes Estatísticos Utilizando Métodos de Seleção de Atributos

Analisando os resultados do teste de *Friedman* da Tabela 6 é possível afirmar que os quatro métodos são diferentes do ponto de vista estatístico para todas as bases de dados. Através da comparação realizada pelo teste *post-hoc*, pudemos observar que, do ponto de vista estatístico, quando comparado ao CIFE, o método avaliado (SD-IM 8T1V) obteve desempenho estatístico superior em oito base de dados, desempenho similar em uma base de dados e desempenho inferior em três bases de dados. Quando comparado ao MIFS, o método avaliado obteve desempenho superior em oito base de dados, similar em uma base e inferior em apenas três bases de dados. Por fim, quando comparado a seleção randômica, o desempenho do método avaliado foi superior em seis bases de dados, similar em duas bases de dados e inferior em apenas quatro bases de dados. De forma geral, os resultados obtidos nas Tabelas 5 e 6 são bastante promissores, uma vez que quando observado os valores de acurácia, o método avaliado obteve desempenho superior aos métodos CIFE e MIFS e desempenho similar a seleção randômica (RFS). Por outro lado, observando os resultados estatísticos, o método passa a ter desempenho superior em comparação a todos os demais, o que corrobora para os bons resultados alcançados com o método avaliado (SD-IM 8T1V).

4.3. Análise Comparativa com Métodos de Extração e Conjunto Original de Dados

Nesta parte final dos resultados, objetivamos verificar o desempenho do método SD-IM em relação a alguns métodos de extração de atributos conhecidos na literatura, PCA e LDA. A tabela 7 apresenta as acúcias dos 3 métodos de redução de dimensionalidade, SD-IM, PCA e LDA, além dos resultados obtidos com a base de dados original, utilizando todo o conjunto de atributos.

Análise Comparativa - Métodos de Extração de Atributos e Conjunto Original de Dados				
Métodos	SD-IM (8T1V)	PCA	LDA	Original
Métricas	Média±Std	Média±Std	Média±Std	Média±Std
B1	77.38±13.54	59.37±10.19	98.18±1.53	64.29±12.67
B2	96.26±4.73	96.06±4.33	99.70±0.67	95.92±3.41
B3	97.18±6.66	90.55±1.67	97.00±1.16	92.12±1.67
B4	94.27±9.12	75.54±4.62	80.41±4.24	77.69±6.04
B5	98.81±2.78	82.54±3.74	91.41±2.03	87.59±2.72
B6	78.00±18.33	84.28±10.97	96.10±4.25	79.01±17.30
B7	76.49±2.37	91.54±5.67	91.17±1.78	92.91±4.52
B8	83.40±4.27	76.84±6.04	99.90±0.18	84.41±3.93
B9	99.95±0.25	62.19±2.55	75.05±6.70	68.47±14.29
B10	73.39±15.64	70.33±11.69	61.10±11.65	78.08±7.71
B11	93.46±5.93	95.84±4.16	99.93±0.12	95.04±4.99
B12	53.47±7.90	51.14±3.22	55.16±3.49	59.863±7.15

Tabela 7. Resultados Utilizando Métodos de Extração e Conjunto Original de Dados

Ao analisar a tabela 7, podemos observar que o PCA obteve o melhor desempenho em nenhuma base de dados. Já a base de dados original obteve o melhor resultado em 3 bases de dados. Além disso, o método avaliado (SD-IM) obteve o melhor desempenho em 4 bases de dados, enquanto o LDA obteve o melhor desempenho em 5 bases de dados. É importante destacar o desempenho obtido pelo LDA, que foi um pouco melhor que o método avaliado. Porém, o método avaliado é uma abordagem de seleção, onde um sub-conjunto de atributos da base de dados original é selecionado. Já o LDA é um extrator de características e cada um dos descritores lineares obtidos representa diferentes combinações de todos os atributos. Desta forma, o LDA produz um sub-conjunto de discriminantes com a combinação de todo o conjunto de atributos, enquanto que o método avaliado produz um sub-conjunto de atributos selecionados. Assim, esta diferença no tipo dos métodos justifica o desempenho obtido pelo LDA, quando comparado ao SD-IM. Na Tabela 8 estão apresentados os resultados dos testes estatísticos, comparando os quatro métodos de redução de dimensionalidade, para todas as bases de dados.

Ao analisar a Tabela 8, podemos observar que, do ponto de vista estatístico, o desempenho do método avaliado foi inferior apenas para o LDA, onde o SD-IM foi superior em 5 e inferior em 7. Podemos, assim, inferir que o modelo proposto apresentou um desempenho competitivo, uma vez que ele quase obteve o mesmo desempenho que um método consagrado como o LDA e que utiliza informação de todo o conjunto de atributos. Além disso, podemos observar que quando comparado ao PCA, o desempenho estatístico foi superior em 9 base de dados, resultado bastante diferente do obtido em relação ao LDA. Assim como o LDA, o PCA também é um extrator de característica que

Testes Estatísticos - Métodos de extração e Conjunto Original de Dados				
Métodos	Friedman	(SD-IM 8T1V) - PCA	(SD-IM 8T1V) - LDA	(SD-IM 8T1V) - Original
Métricas	p-value	p-value	p-value	p-value
B1	3.92E-113	<	>(*)	<
B2	4.01E-38	<	>(*)	<
B3	3.92E-113	<	<	<
B4	1.55E-73	<	<	<
B5	1.30E-136	<	<	<
B6	7.88E-81	>(*)	>(*)	=
B7	6.54E-108	>(*)	>(*)	>(*)
B8	4.70E-100	<	>(*)	>(*)
B9	9.01E-122	<	<	<
B10	7.04E-31	<	<	>(*)
B11	4.58E-93	>(*)	>(*)	>(*)
B12	1.18E-44	<	>(*)	>(*)

Tabela 8. Resultados dos Testes Estatísticos Utilizando Métodos de Extração de Atributos e Conjunto Original de Dados

utiliza informações de todo o conjunto de características. Mesmo assim, o método avaliado conseguiu obter melhor desempenho que o PCA. Finalmente, em relação ao conjunto de dados original, o método SD-IM foi superior em 6 bases de dados, similar/equivalente em uma base e inferior em apenas 5 bases de dados.

5. Conclusões

Este artigo apresentou um estudo exploratório de um método dinâmico de seleção de atributos que utiliza um algoritmo de agrupamento com medidas de informação mútua durante seu processo de seleção. O principal objetivo deste método dinâmico é tornar o processo de seleção de atributos mais eficiente através da escolha de atributos que seja o melhor para um grupo de instancias, em vez de apenas um conjunto de atributos para todo o conjunto de teste.

Para a realização do estudo exploratório, uma análise empírica foi realizada. Nesta análise, verificamos o impacto da proporção conjunto validação/teste tinha no desempenho do método SD-IM. Além disso, fizemos duas análises comparativas, onde a primeira comparou o desempenho do SD-IM com outros métodos de seleção baseados em informação mútua e a segunda comparou com o desempenho de métodos conhecidos na literatura.

Como resultado desta análise, podemos concluir que o método SD-IM teve um desempenho bem competitivo, uma vez que o mesmo obteve melhor desempenho que a maioria dos outros métodos. A única exceção foi o LDA, cujo desempenho foi um pouco inferior. Porém, como já mencionado, ao contrário do método SD-IM, o LDA é um extrator de características que ao final produz um sub-conjunto de discriminantes que possuem a combinação de todo o conjunto de atributos. Por outro lado, o SD-IM utiliza apenas informações do sub-conjunto de atributos selecionados. Essa diferença de funcionamento justifica o desempenho superior do LDA em relação ao SD-IM.

6. Agradecimentos

Os autores gostariam de agradecer o suporte financeiro oferecido pelo CNPq e pela PROPESQ/UFRN.

Referências

- Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *Trans. Neur. Netw.*, 5(4):537–550.
- Friedman, M. (1937). The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32(200):675–701.
- Gibbons, J. and Chakraborti, S. (2003). *Nonparametric Statistical Inference*. Statistics, textbooks and monographs. Marcel Dekker Incorporated.
- Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182.
- Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Jesus, J. K. L., Canuto, A. M. P., and Araújo, D. S. A. (2017a). Estudo exploratório sobre uma abordagem de seleção dinâmica de atributos no contexto de classificação de padrões. In *4^o Workshop de Pesquisa Científica do PPgEEC (WPC)*.
- Jesus, J. K. L., Canuto, A. M. P., and Araújo, D. S. A. (2017b). A feature selection approach based on information theory for classification tasks. In *2017 International Conference on Artificial Neural Networks (ICANN)*.
- Jolliffe, I. T. (1986). *Principal Component Analysis*. Springer-Verlag, Berlin; New York.
- Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):881–892.
- Li, J., Cheng, K., Wang, S., Morstatter, F., Robert, T., Tang, J., and Liu, H. (2016). Feature selection: A data perspective. *arXiv:1601.07996*.
- Lichman, M. (2013). UCI machine learning repository.
- Lin, D. and Tang, X. (2006). *Conditional Infomax Learning: An Integrated Framework for Feature Extraction and Fusion*, pages 68–82. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Liu, H. and Motoda, H. (2007). *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC.
- McLachlan, G. (2004). *Discriminant Analysis and Statistical Pattern Recognition*. Wiley Series in Probability and Statistics. Wiley.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.
- Nunes, R. O., Dantas, C. A., Canuto, A. M. P., and Xavier-Junior, J. C. (2016). An unsupervised-based dynamic feature selection for classification tasks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 4213–4220.
- Seville, B. G. (2004). BIGS bioinformatics research group of seville repository.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27.

Método Supervisionado para Categorização Automática de Lançamentos Financeiros

Cleviton V. F. Monteiro¹, Antonio Jorge Delgado¹, Rinaldo Lima¹, Edvan Soares¹,

¹Departamento de Estatística e Informática (DEINFO) – Universidade Federal Rural de Pernambuco (UFRPE)

Rua Dom Manoel de Medeiros – 52171-900 – Recife – PE – Brazil
{cleviton, ajffdelgado, edvanjsj}@gmail.com, rinaldo.jose@ufrpe.br

Abstract. *The financial management is one of the management disciplines that deserves more attention and effort. On the other hand, when it is poorly managed, it may cause financial problems that, in turn, causes the company's bankruptcy. To mitigate such a problem, this work aims to reduce the managers' effort at the task of organizing financial records by automatic classifying them into the correct financial accounts. The results achieved with the Logistic Regression classifier were the most promising. The F1-measure and accuracy scores were, respectively, 92,9 and 92,68. These results are encouraging because they suggest that it is viable to automate the manual task currently performed by company managers.*

Resumo. *A gestão financeira em empresas é uma das áreas que mais requer esforço e atenção, visto que, quando mal administrada, pode levar a problemas como a insuficiência de recursos que, por sua vez, pode levar a falência da empresa. Para mitigar tal problema, este trabalho visa reduzir o esforço dos gestores de empresas na tarefa de organização das informações de lançamentos bancários através da classificação automática deles em suas devidas contas contábeis. Os resultados obtidos pelo classificador de regressão logística obteve o melhor desempenho com os scores de F1-measure e acurácia de 92,9 e 92,68, respectivamente. Tais resultados são encorajadores dado que grande parte do esforço na entrada manual dos lançamentos poderão ser realizados automaticamente, fornecendo uma alternativa viável para auxiliar os gestores na tarefa em questão.*

1. Introdução

As pequenas e médias empresas brasileiras têm sofrido com dificuldades para se manterem ativas por um longo período. Dados do Sebrae [SEBRAE, 2015] apontam que em torno de 24% delas não sobrevivem ao segundo ano de atividade. Dentre os motivos que levam ao fim precoce dessas empresas, pode-se destacar a falta de controle financeiro ou um controle financeiro precário, principalmente do fluxo de caixa [SEBRAE, 2004; SEBRAE, 2014]. Tal problema gera insuficiência de recursos que, por sua vez, pode levar a falência principalmente de pequenas empresas. Para que isso seja evitado, faz-se necessário um planejamento financeiro que conduza a um planejamento estratégico visando o sucesso do negócio [Ferreira et al., 2012; Souza e Qualharini, 2007].

Uma forma de mitigar o problema acima mencionado seria reduzir ou eliminar barreiras que dificultem o uso de sistemas de gestão financeira pelas empresas. Dessa forma, para realizar um controle financeiro eficaz, os gestores precisam cadastrar as vendas realizadas, previsões de pagamentos e recebimento, além de analisar o extrato bancário de uma ou mais contas correntes para identificar essas operações de entrada e saída, e categorizá-las corretamente. Observa-se que é necessário um esforço considerável do gestor e

dos seus assistentes na execução dessas atividades. Porém, tais empresas costumam ter quadros reduzidos de funcionários e muitas vezes acabam negligenciando o monitoramento do controle financeiro.

Sendo assim, observou-se a oportunidade de auxiliar os gestores com o uso de aprendizagem supervisionada para que, a partir dos lançamentos dos extratos fornecidos pelos bancos, seja possível categorizar automaticamente cada lançamento nas contas contábeis definidas pela empresa em questão. Tais contas contábeis são usadas como rótulos na classificação (atributo alvo) e estão relacionadas às receitas e despesas das empresas, tais como: salário, aluguel, energia, pagamento de fornecedores, dentre outras. De posse dos lançamentos devidamente categorizadas, os gestores passam a ter mais elementos para a tomada de decisão.

O objetivo deste artigo é avaliar algoritmos de classificação de texto e comparar os resultados da aplicação de cada um deles à categorização de lançamentos financeiros em contas contábeis. O desafio aqui reside no fato que, para conseguir atingir um alto desempenho (em termos de acurácia, por exemplo) na classificação automática de tais lançamentos em suas devidas contas contábeis, reside em dois fatores cruciais: a restrita quantidade de informação, isto é, curtas descrições disponíveis para geração de atributos (*features*) dos lançamentos bancários fornecidos pelas instituições bancárias; e um número grande e variável de classes (contas contábeis) possíveis.

O restante deste trabalho está organizado da seguinte forma: na Seção 2 apresentamos os algoritmos supervisionados de aprendizagem de máquina selecionados por este estudo. A Seção 3 apresenta os trabalhos relacionados, enquanto na Seção 4 detalhamos a proposta. A Seção 5 discute os resultados experimentais. Seção 6 finaliza o artigo com as conclusões e perspectivas de trabalho futuro.

2. Algoritmos Supervisionados para Classificação de Textos

Essa seção visa apresentar, de forma sucinta, os algoritmos de classificação de texto selecionados neste trabalho: Support Vector Machines (SVM) [Cortes e Vapnik, 1995], Naive Bayes (NB) [Wu, 2007], C4.5 [Quinlan, 1993], Decision List [Frank et al., 2005] e Regressão Logística (RL) [Wu, 2007]. Esses algoritmos são os mais utilizados em mineração de textos devido ao alto desempenho obtidos em várias tarefas típicas, incluindo classificação de documentos e análise de sentimentos [Wang e Manning, 2012]. Os algoritmos de classificação têm o objetivo de realizar tarefas *preditivas*, onde eles constroem, numa fase de aprendizado, um modelo capaz de identificar os padrões de exemplos previamente anotados com sua classe ou categoria. Os modelos assim aprendidos podem ser usados para atribuir uma classe aos novos exemplos de entrada numa fase de aplicação do modelo aprendido [Al Hasan, 2006].

O algoritmo SVM é um classificador binário não-probabilístico que encontra uma fronteira de decisão que separa os dados de entrada em duas classes e, ao mesmo tempo, maximiza a distância entre os pontos com maior proximidade em cada uma das classes [Meyer, 2001].

Já o algoritmo Naive Bayes consiste em um algoritmo de classificação probabilística baseada no teorema de Bayes. Esse classificador assume a independência entre os atributos que descrevem os exemplos [Leung, 2007]. Uma de suas vantagens é a necessidade de uma quantidade pequena de dados para o treinamento e sua boa eficiência computacional na fase de treinamento.

O algoritmo C4.5 é um classificador capaz de criar regras de classificação derivadas de árvores de decisão obtidas do conjunto de dados de entrada [Quinlan, 1993]. As árvores de decisão são construídas através de um processo de particionamento recursivo do conjunto de dados de entrada por meio de testes dos valores de seus atributos.

O algoritmo de lista de decisão constrói uma árvore de decisão parcial (C4.5) em cada iteração e converte o caminho da melhor folha encontrada como uma regra [Frank e Witten, 1998].

Por fim, o algoritmo de regressão logística (RL) gera modelos de regressão linear baseado no algoritmo de boosting *LogitBoost* que usa funções de regressão simples como classificador base para ajustar os modelos logísticos [Sumner et al., 2005].

3. Trabalhos Relacionados

Esse trabalho visa aplicar aprendizagem de máquina supervisionada para identificar e categorizar automaticamente os lançamentos bancários de uma empresa. O problema de classificação automática de textos curtos tem sido abordado na literatura [Lakshmi, 2010; Sajedi, 2016].

Phan e colegas [Phan et al., 2008] desenvolveram um framework para classificação de textos curtos e esparsos, onde um conjunto de dados externos é obtido e integrado ao conjunto de treinamento. A ideia deles consiste em construir um classificador baseado tanto em um pequeno conjunto de dados rotulados para treinamento quanto em um conjunto rico de dados externos. Esse classificador foi aplicado, por exemplo, para a classificação de resumos de artigos médicos em cinco categorias de doenças.

Lakshmi (2010) utilizou o algoritmo de aprendizagem supervisionado no reconhecimento de padrões que identificam se um e-mail é um spam ou não. Ele empregou os classificadores SVM e Naive Bayes para realizar a classificação dos spams.

Sajedi (2016) usou o algoritmo Naive Bayes para identificação de spam em mensagens SMS. O que distingue esses dois últimos trabalhos é o fato que mensagens SMS são relativamente bem menores em tamanho do que o conteúdo de mensagens de e-mails. Este fator (texto menor) com certeza introduz desafios e pode influenciar diretamente na acurácia dos modelos gerados pelos algoritmos de aprendizagem supervisionada [Sajedi, 2016]. Os resultados alcançados pelos autores correspondem a uma acurácia de 98,20 com a utilização do algoritmo Naive Bayes e de 97,64 com o SVM.

Outra aplicação que envolveu a classificação de textos curtos pode ser encontrada em Pang et al. (2002) que classifica opiniões sobre filmes em sentimentos positivo e negativo. Foram utilizados os algoritmos Naive Bayes e SVM, tendo este último obtido os melhores resultados em relação ao NB e sendo comparáveis as classificações geradas por humanos.

Roberts et al. (2012) também lidou com o problema de classificação de textos curtos, em particular, os autores criaram um classificador que categoriza sentimentos em tweets, detectando emoções como amor, surpresa, raiva, e etc. Eles empregaram o algoritmo SVM para detectar sete (7) emoções no total, usando o software de mineração de dados WEKA [Frank et al., 2005].

Um sistema de lançamentos financeiros que tratou do problema de classificação de lançamentos financeiros que mais se assemelha a solução proposta neste artigo é o “Minhas Economias” (2017) que consiste num sistema Web gratuito para categorizar gastos e despesas pessoais. Esse sistema categoriza os lançamentos através do uso de palavras-chave obtidas da descrição do lançamento, por isso, ele apresenta a séria limitação de que o gestor precisa

digitar a descrição do lançamento e gerenciar as palavras-chave manualmente. Em outras palavras, não é possível importar automaticamente os lançamentos de um extrato ou de categorizá-los automaticamente como feito pelo sistema aqui proposto.

4. Método Proposto

O presente estudo adota o processo computacional de descoberta de conhecimento chamado *Knowledge Discovery in Databases* (KDD) em inglês [Imielinski e Mannila, 1996] que se caracteriza por estruturar o processo de descoberta de conhecimento nas seguintes fases: *seleção, pré-processamento, transformação, mineração de dados e interpretação e avaliação* do conhecimento descoberto. Nas seções subsequentes, será descrito em que consiste cada etapa deste processo e como foram customizadas neste trabalho.

4.1. Seleção

Essa etapa consiste em escolher os conjuntos de dados (datasets) que possuem os atributos e exemplos (registros) dos quais se deseja fazer a descoberta de conhecimento. A escolha desses dados foi feita por um especialista da área de gestão financeira.

Mais precisamente, o gestor teve a tarefa de obter os extratos bancários de três contas correntes, em formato texto, pertencentes a duas empresas. Os extratos bancários são formados pelos seguintes atributos: *número da agência e conta, nome da empresa, data do lançamento, descrição, detalhamento, número do documento e valor*. A extração dos conteúdos (textual) desses atributos foi realizada por um módulo desenvolvido em Python. A partir dos extratos foi criada uma planilha eletrônica para cada conta corrente das empresas (Figura 1).

Juntamente com os extratos, também foram recebidos os registros financeiros do sistema de gestão das empresas envolvidas para que fosse possível categorizar as informações dos extratos. Sendo assim, para cada lançamento do extrato, foi determinada a *classe-alvo* dos lançamentos (coluna *categoria*, na Figura 1) de acordo com o conjunto de contas contábeis de cada empresa. Importante destacar que cada empresa tem seu conjunto próprio de contas contábeis. Nessa etapa, o gestor identificou alguns erros de classificação no sistema de gestão e realizou as devidas correções de acordo com as recomendações das empresas.

Data de compensação	Descrição na conta corrente	Detalhamento	Documento	Valor	Categoria
18/12/2015	474 Transferência Agendada	17/12 0232 24819-3 ELARALDO COLOIA	23,200,000,024,819	-1043	13º Salário
18/12/2015	474 Transferência Agendada	17/12 1814 29640-6 BRUNO RAMOS C	181,400,000,029,640	-412	13º Salário
18/12/2015	474 Transferência Agendada	17/12 1833 25866-0 ANTONIO JOSE	183,300,000,025,866	-995	13º Salário
18/12/2015	474 Transferência Agendada	17/12 1850 13503-8 LARA LUCIA XAV	185,000,000,013,503	-133	13º Salário
18/12/2015	474 Transferência Agendada	17/12 3613 22477-4 RENATO CELSO S	361,300,000,022,477	-894	13º Salário
18/12/2015	474 Transferência Agendada	17/12 3613 44149-X JAMESSON RICARDO	361,300,000,044,149	-995	13º Salário
18/12/2015	474 Transferência Agendada	17/12 3613 48388-5 UBIRATAN CLEY	361,300,000,048,388	-995	13º Salário
18/12/2015	474 Transferência Agendada	17/12 3613 56685-3 RIVALDO GONCALV	361,300,000,056,685	-996	13º Salário
20/11/2015	470 Transferência on line	20/11 4357 58893-8 ITALO JORGE DA	664,357,000,058,893	-3209.54	13º Salário
29/09/2015	470 Transferência on line	29/09 1509 15539-X JOSÉ SANTO	661,509,000,015,539	-93.87	13º Salário
30/11/2015	474 Transferência Agendada	30/11 0232 24819-3 ELARALDO COLOIA	23,200,000,024,819	-1392	13º Salário
30/11/2015	474 Transferência Agendada	30/11 1814 29640-6 BRUNO RAMOS C	181,400,000,029,640	-373	13º Salário
30/11/2015	474 Transferência Agendada	30/11 1833 25866-0 ANTONIO JOSE	183,300,000,025,866	-1319	13º Salário
30/11/2015	474 Transferência Agendada	30/11 1850 13503-8 LARA LUCIA XAV	185,000,000,013,503	-75	13º Salário
30/11/2015	474 Transferência Agendada	30/11 3613 22477-4 RENATO CELSO S	361,300,000,022,477	-1099	13º Salário
30/11/2015	474 Transferência Agendada	30/11 3613 44149-X JAMESSON RICARDO	361,300,000,044,149	-1319	13º Salário
30/11/2015	474 Transferência Agendada	30/11 3613 48388-5 UBIRATAN CLEY	361,300,000,048,388	-1318	13º Salário
30/11/2015	474 Transferência Agendada	30/11 3613 56685-3 RIVALDO GONCALV	361,300,000,056,685	-1319	13º Salário

Figura1. Extrato da planilha eletrônica criada.

A Tabela 1 indica a quantidade total de registros (linhas) por conta corrente de cada empresa, a número de classes-alvo (categorias), o tamanho médio da coluna “descrição” em

número de caracteres, assim como o tamanho médio da coluna “detalhamento”. Nota-se que a quantidade de linhas dos datasets é pequena e os textos são bem curtos, refletindo os escarços dados dos extratos bancários das empresas.

Por fim, para enriquecer os dados dos extratos, incorporou-se outros dados obtidos do sistema de gestão das empresas. As informações obtidas e integradas estão relacionadas à identificação do nome de funcionários, clientes ou fornecedores em um dado lançamento. Esse processo foi realizado manualmente e pretende-se automatizá-lo em trabalho futuro.

Tabela 1. Estatísticas básicas dos datasets das empresas

Datasets	Total de Registros	Qtd. de classes	Tamanho médio da coluna descrição (# carac.)	Desvio padrão da coluna descrição	Tamanho médio da coluna detalhamento (# carac.)	Desvio padrão da coluna detalhamento
Empresa 1(E1)	350	30	22,64	4,83	21,08	13,30
Empresa2-Conta1(E2C1)	524	55	22,48	3,92	10,87	11,30
Empresa2-Conta2(E2C2)	396	15	23,86	2,10	17,06	14,57

4.2. Pré-processamento

Após a extração e classificação manual dos registros da planilha, foram geradas várias informações, com o objetivo de estudar a distribuição dos exemplos em relação as suas classes e assim poder introduzir as correções necessárias. Tal tarefa foi implementada por um outro módulo de nossa solução usando a biblioteca Matplotlib disponível em Python para geração de gráficos por meio de scripts interativos [Hunter, 2007].

A partir desse script, foram gerados histogramas e outras medidas estatísticas por meio de gráficos bloxplots para analisar as medidas de tendência central, dispersão, e percentis das classes de contas contábeis das empresas. A Figura 2 exibe a frequência de lançamentos financeiros por classe-alvo da Empresa 1 (E1). Observa-se que há um desbalanceamento da quantidade de lançamentos bastante acentuado. Assim, algumas classes, como "Transferência", "Salário", "Venda de serviços" e "Despesas bancárias", são bastante frequentes, enquanto outras como "Água" e "Ajuste de caixa" são bem menos frequentes. Nesse caso, foi preciso fazer um balanceamento das categorias menos frequentes. Mais precisamente, para todas as classe com menos de 4 exemplos foram então replicados, de forma aleatória, para exatamente 4 exemplos. Chegou-se a tal número mínimo de representative de uma classe nos datasets de entrada, após experimentos iniciais que mostraram que os algoritmos de classificação já conseguiam inferir um modelo se as classes tivessem pelo menos 4 exemplos. Este tipo de pré-processamento foi o único que gerou mudança no dataset, mantendo-se todas as demais características originais deles. Outras análises foram realizadas para se conhecer a distribuição dos dados e escolher as estratégias de transformação de dados descritas na etapa seguinte.

4.3. Transformação dos dados

Nesta etapa, apenas os três atributos textuais “descrição”, “detalhamento” e “no. documento dos lançamentos bancários nos extratos são segmentá-los em termos ou *tokens* e usados para gerar um único vetor unidimensional contendo todos os *tokens* distintos encontrados em todos os lançamentos de um mesmo dataset.

Tal abordagem de representação de atributos textuais é também chamada de *Bag-of-Words* (BOW) sendo, de fato, uma das mais simples, porém efetivas técnicas de representação de documentos textuais usados em algoritmos de classificação supervisionada [Feldman e Sanger, 2006]. Usando esta técnica, cada lançamento torna-se um vetor denotando assiuma instância de um vetor em um espaço vetorial de representação para todos os lançamentos [Roberts et al., 2012].

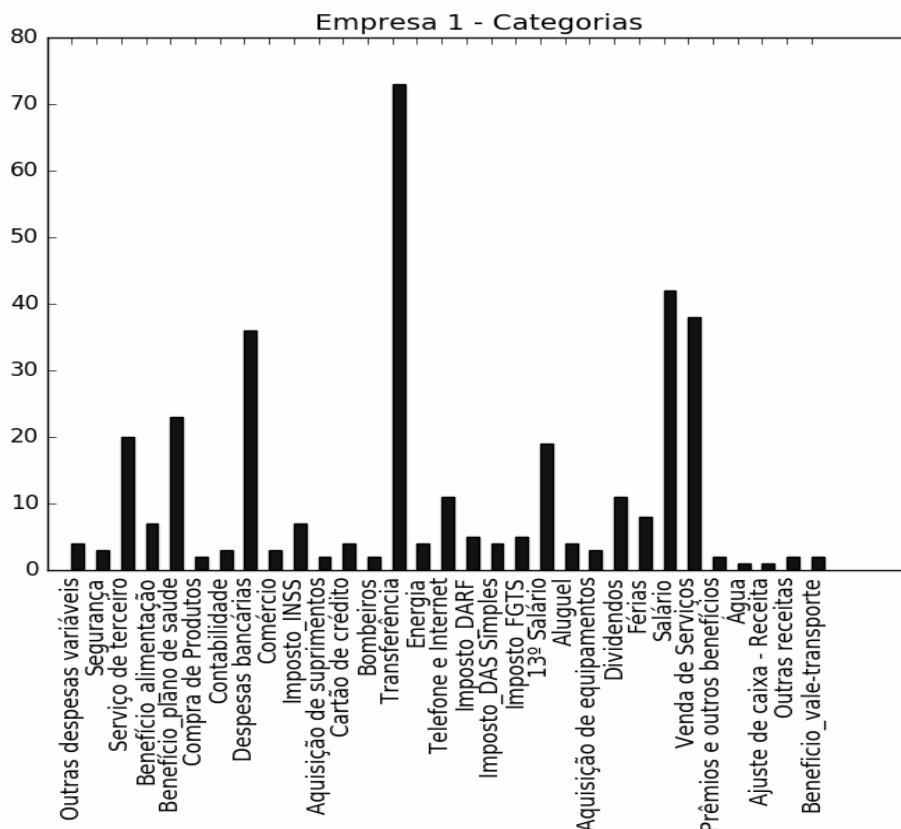


Figura 2. Distribuição das frequências das classes do conjunto de dados da Empresa 1

Ao final do processo de tokenização acima, todos os *tokens* distintos encontrados são usados para definir colunas em um vetor unidimensional capaz de representar qualquer lançamento presente no dataset. Este vetor unidimensional, assim gerado, é também chamado de *vocabulário* [Feldman e Sanger, 2006]. Por exemplo, ao ser analisado o texto da coluna “Descrição na conta corrente” de um lançamento, os *tokens* gerados a partir dela seriam utilizados para criar novas colunas identificadas pela prefixo “dsc_” seguido pela descrição do *token*. Dessa forma, podem ser gerados muitos outros atributos como novas colunas nomeadas segundo o formato *desc token*. Como resultado, novos atributos (colunas na planilha), passam a ser identificada por *dsc_token*, onde a parte “token” corresponde ao segmento de texto extraído do campo “Descrição na conta corrente”. Por exemplo, se em um dos lançamentos, o texto do campo “descrição” for “474 Transferência Agendada” (1ª. Linha da Figura 3a), este lançamento será tokenizado originando três novos tokens: “474”, “Transferência” e “Agendada” que, por sua vez, originariam novos atributos no dataset de entrada: “dsc 474”, “dsc transferencia” e “dsc agendada”. Todos os tokens textuais são convertidos em letras minúsculas e removidos os acentos.

Após a etapa de tokenização anterior, os novos atributos gerados são preenchidos com o valor 1, indicando se, em um lançamento particular, existem os tokens em questão (Figura 3b); e preenchidos com 0, se o referido *token* não está presente na referida linha. Para ilustrar este processo, considere que se deseja transformar um lançamento contendo o texto “470

“Transferência on line” na coluna de descrição. Logo, as colunas *dsc_474* e *dsc_agendada* (colunas geradas pelo exemplo anterior) teriam o valor 0, já que não possuem esses tokens no lançamento atual, e apenas a coluna *dsc_transferencia* receberia o valor 1, pois o exemplo atual contém este token (última linha da Figura 3b).

A representação binária dos datasets foi adotada, em detrimento de outras tais como o TF/IDF [Feldman e Sanger, 2006], pelo fato de que os algoritmos de classificação selecionados apresentam restrições quanto a forma de entrada dos vetores de treinamento, por exemplo, o classificador SVM que assume que os valores estejam normalizados, isto é, entre 0 e 1 ou -1 e 1. Visando também deixar aberta a opção de se testar outros classificadores, tais como redes neurais que igualmente esperam valores normalizados nos datasets de treinamento. Portanto, a representação binária então tornou-se a mais indicada para os fins de nosso estudo.

Descrição na conta corrente	dsc_474	dsc_Transferência	dsc_470	dsc_766
474 Transferência Agendada	1	1	0	0
474 Transferência Agendada	1	1	0	0
470 Transferência on line	0	1	1	0
470 Transferência on line	0	1	1	0
766 Doc Devolvida	0	0	0	1
470 Transferência on line	0	1	1	0

(a)

(b)

Figura 3. Exemplo de uma coluna e sua conversão em representação binária

Devido aos seus tipos, os campos de data e valor da planilha foram igualmente representados de forma binária, porém realizando-se uma discretização em vários intervalos antes. Para as datas, foram criadas colunas que indicam se o valor do lançamento está nos 5 primeiros dias do mês, do dia 6 ao 10, do 11 ao 15, e assim por diante. De igual forma, foram criados outros intervalos de 10 em 10 dias e de 15 em 15 para este campo. Já para o campo de valor, devido à sua maior amplitude dos valores, foram adotados a discretização em 100 partes iguais, gerando intervalos de 10 em 10. Ao final da etapa de transformação de dados, as planilhas originais das contas correntes das empresas que possuíam originalmente 6 campos, passaram a ter 1226, 1548 e 915; para os datasets E1, E2C1, E2C2, respectivamente.

4.4. Mineração de dados

Na etapa de mineração dos dados, foram selecionados os algoritmos de aprendizagem de máquina supervisionada descritos na Seção 2 conforme implementação disponível no software de mineração de dados WEKA [Frank et al., 2005]. Em particular, foram usados os seguintes algoritmos: Naive Bayes, *Simple Logistic* (Regressão Logística), SMO (SVM), J48 (árvore de decisão) e *PART* (lista de decisão). Esses algoritmos foram selecionados porque eles estão entre os que tem obtido um bom desempenho em várias tarefas de classificação de textos de acordo com [Feldman e Sanger, 2006; Lakshmi e Rada, 2010; Pang et al., 2002; Leung, 2007]. Para o algoritmo SVM, foi usado o kernel linear, uma vez que os kernels polinomiais e RBF (*Radial basis Function*) não apresentaram melhores resultados do que o kernel linear em experimentos preliminares usando avalidos nos mesmos datasets.

Para todos os algoritmos foram usados seus parâmetros padrão (*default*) e foi utilizada validação cruzada (*cross-validation*) com 10 *folds* visto que tal divisão dos dados em treinamento e teste e permite obter uma estimativa mais confiável do desempenho real dos modelos gerados pelos classificadores selecionados. A técnica de validação cruzada é uma

estratégia de avaliação na qual os dados são divididos em k partes, também chamados de *folds*, separando-se $k-1$ destas partes para serem usadas no treinamento do algoritmo e a parte restante para se avaliar ou estimar o desempenho do classificador [Arlot et al., 2010].

4.5. Interpretação e Avaliação

Para interpretar os resultados obtidos ao final da etapa anterior, foram levadas em consideração as métricas geradas pelo WEKA, incluindo precisão (*precision*), cobertura (*recall*), *F-measure*, acurácia (*accuracy*), e a matriz de confusão de cada classificador e a opinião dos especialistas no domínio. As principais conclusões dessa etapa de avaliação e interpretação dos resultados são discutidas na seção a seguir.

5. Resultados Experimentais

A Tabela 2 sumariza os resultados obtidos pelos classificadores para cada dataset das empresas em termos das métricas de precisão (P), cobertura (C), F1-measure ($F1$) e acurácia (A). Precisão denota a proporção das instâncias pertencentes a uma classe A que foram classificadas corretamente como sendo da Classe A .

A cobertura é a proporção de instâncias classificadas em uma determinada classe A entre todos que realmente pertencem a essa categoria A , ou seja, o quanto os representantes de A foram capturados pelo classificador. A *F-Measure* é definida como a média harmônica entre as métricas de Precisão e a Cobertura dada pela fórmula $(2 * P * R / (P + R))$. Já a Acurácia é a porcentagem das instâncias de todas as classes que foram classificadas corretamente [Bouckaert et al. 2010].

Tabela 2. Resultados dos classificadores nos 3 datasets selecionados
Classificadores: Naive Bayes (NB), Regressão Logística (RL), Support Vector Machines (SVM), Árvore de Decisão (J48), Decision List (PART). Datasets: Empresa 1 (E1), Empresa 2/ Conta 1 (E2C1), Empresa 2/Conta 2 (E2C2)

	Dataset	NB	RL	SVM	J48	PART
Precisão	E1	86,40	93,10	92,70	88,60	85,80
	E2C1	63,90	80,40	74,70	72,80	73,00
	E2C2	85,60	88,80	87,00	86,20	87,20
Cobertura	E1	81,60	92,70	92,40	89,70	86,40
	E2C1	60,40	80,20	76,10	74,70	73,90
	E2C2	83,50	88,20	86,90	86,50	86,90
F1-Measure	E1	82,40	92,80	92,40	88,90	85,80
	E2C1	59,70	79,70	74,70	73,30	72,50
	E2C2	84,10	88,30	86,80	85,90	86,90
Acurácia	E1	81,57	92,68	92,41	89,70	86,45
	E2C1	60,35	80,18	76,14	74,74	73,86
	E2C2	83,50	88,18	86,95	86,45	86,95

Observa-se que o classificador RL teve desempenho superior aos demais em todos os datasets avaliados. Ele foi acompanhado de perto pelo SVM em termos de F1-measure e acurácia nos datasets E1 e E2C1; enquanto que no dataset E2C2, o RL teve desempenho comparável com os classificadores baseados em árvore de decisão J48 e PART. O classificador NB foi o que apresentou os mais baixos escores na avaliação.

De fato, foi realizado o teste Wilcoxon *signed-rank* [Gibbons & Chakraorti, 2011] para avaliar a significância estatística entre as médias dos scores de F1 entre todos os classificadores. Os resultados dos teste comprovam haver uma diferença estatística significativa no nível de confiança de 95% ($\alpha = 0,05$) do desempenho entre o classificador RL e os demais classificadores em todos os datasets, seguido pelo SVM que apresenta diferença estatística significativa em relação ao NB, J48 e PART nos datasets E1 e E2C1. Porém, para o dataset E2C2, o classificador SVM e os algoritmos baseados em árvore de decisão foram igualmente eficientes.

Outro conhecimento extraído importante para o estudo, é que, entre todos os datasets avaliados, o E2C1 foi o mais difícil, enquanto que o E1 foi o mais fácil. Considerando que todos os datasets possuem características gerais similares, isto é, mesmo número de colunas iniciais, porém derivados de populações diferentes, uma possível explicação para o nível de dificuldade encontrado pelos classificadores se deve ao fato de que o dataset E2C1 é o que possui o maior número de classes entre todos os datasets avaliados. Além disso, o tamanho médio das informações das colunas mais importantes (descrição e detalhamento) é menor em relação aos outros datasets. Assim, os classificadores tem menos features disponíveis para usando no processo de construção dos modelos.

Seguindo este mesmo raciocínio, a diferença do tamanho da coluna de detalhamento é a segunda menor entre os datasets, o que explicaria que o dataset E2C2 ser o segundo dataset que apresentou mais dificuldade para os classificadores.

De modo geral, pode-se concluir que, a taxa de acertos (acurácia) média dos algoritmos avaliados tem atingido resultados encorajadores, com uma média de 75,14, 87,01, 85,16, 83,45 e 82,42, para os classificadores NB, RL, SVM, J48, e PART, respectivamente. Isso implica que, na aplicação prática desses classificadores na categorização automática de lançamentos financeiros, a mais alta acurácia atingida significa que quantidade de lançamentos que os gestores precisariam alterar manualmente seria de 25% (no pior caso) e menos de 13% (no melhor caso).

Para analisar detalhadamente a quantidade de falso-positivos entre todas as classes, apresentamos a matriz de confusão do melhor modelo gerado pelo classificador RL avaliado no dataset da Empresa E1 (Figura 4). Sendo apenas essa apresentada por falta de espaço.

A Figura 4 mostra que cada célula da matriz de confusão faz um cruzamento da quantidade de classificações realizadas por um certo classificador na classe da coluna contra a classificação correta da linha. Dessa forma, a quantidade de classificações corretas encontra-se nas células da diagonal principal da matriz, e as incorretas, fora dessa diagonal. Ao todo foram realizadas 342 classificações corretas e 27 incorretas para o modelo gerado pelo classificador RL no dataset E1.

Segundo esta matriz, mais da metade das 30 classes no dataset E1 obtiveram um resultado perfeito (acurácia 100%), mesmo para classes com relativamente poucos exemplos, como é o caso para as classes "aluguel", "imposto_inss", "imposto_das", entre outras. Por outro lado, apenas uma (1) classe, que é a de despesas variáveis (*desp_var*), teve o pior desempenho entre todas (0% de acurácia. A razão para isso é que tal categoria é realmente, por definição, a menos confiável entre todas, pois ela agrega várias outras despesas sem um

detalhamento ou descrição específicas para cada uma delas. Devido a este fato, mais esforço vai ser empenhado para melhorar o desempenho para esta classe. Mais precisamente, informações complementares deverão ser acrescentadas ao dataset para a geração de features adicionais que ajudem na identificação dessa classe.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac	ad	<-- classified as		
17	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0		a = 13_sal	
0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		b = agua
0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		c = aj_caixa
0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		d = alug
0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		e = ag_eq
0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		f = ag_sup
0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		g = ben_ali
0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		h = ben_plano
0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		i = ben_vale
0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		j = bomb
0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		k = c_cred
0	0	0	0	0	0	0	0	0	0	3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		l = comer
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		m = compra
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0		n = contab
0	0	0	0	0	0	0	0	0	0	0	0	0	0	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		o = desp_banc
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0		p = desp_var
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0		q = divid
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	1	0	0	0	0		r = ener
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0		s = ferias
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0		t = imp_darf
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0		u = imp_das
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0		v = imp_fots
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0		w = imp_inss
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0		x = out_rec
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41	0	0	0	0		y = sal
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0	0	0		z = segur
0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	14	1	0	0		aa = serv_terc
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	7	0	0		ab = tel_int
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	72	0	0		ac = transf
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	35		ad = venda_serv

Figura 4. Matriz de confusão do algoritmo RL para a conta da empresa 1 (E1)

6. Conclusões e Trabalhos Futuros

A motivação para o desenvolvimento deste trabalho vem da oportunidade de ajudar gestores de pequenas e médias empresas a conseguirem implantar uma gestão financeira com menor esforço. Em outras palavras, para que os gestores possam tomar decisões com base em informações financeiras corretamente categorizadas, sem que eles realizem muito esforço na organização dos lançamentos financeiros. Ao demandar menor esforço, espera-se que a gestão financeira seja menos negligenciada pelas empresas.

Sendo assim, o objetivo deste trabalho foi avaliar diferentes modelos de classificação de texto para a categorização automática de lançamentos financeiros em suas respectivas contas contábeis. Os resultados mostraram que o classificador RL apresentou as melhores resultados com relação a todas as métricas analisadas, seguido de perto pelo SVM.

De um ponto de vista mais geral, os resultados alcançados em torno, ou mesmo maiores que 80% na métrica de acurácia sugere que a grande maioria das predições feitas pelo modelo RL estão corretas, ou seja, menos de 20% dos lançamentos, no pior caso (E2C1), e 7%, no melhor caso (E1), precisariam ser alterados pelo gestor. Na prática, é possível que mais do que 7% dos lançamentos categorizados pelos gestores possuam erros de classificação. Isso nos leva a hipotetizar que a classificação automática realizada pelo modelo pode até mesmo ser mais precisa do que a realizada por um humano.

Destacamos que os resultados apresentados precisam ser analisados levando em consideração que apenas 3 contas bancárias foram utilizadas na categorização e que isso representa uma limitação do trabalho. Tal quantidade de contas se deve ao fato de ser difícil obter informações dessa natureza para realizar o estudo, visto que a grande maioria das empresas consultadas não quiseram repassar seus lançamentos financeiros e contas contábeis.

Além disso, por questões de contrato de confidencialidade com as empresas, os datasets não podem ser disponibilizados publicamente.

No caminho de aprofundar esse estudo, pretende-se coletar métricas da classificação dos lançamentos pelos gestores (humanos) para comparar com os resultados da categorização automática. Além disso, será preciso elaborar a arquitetura de um framework que automatize todas as etapas do método proposto: desde a obtenção dos lançamentos financeiros, a geração dos modelos de categorização automática, e a integração destes últimos ao sistema de gestão das empresas.

Além disso, esse framework irá possibilitar a coleta de novos dados e contas contábeis de empresas, visando a evolução do próprio framework em relação a geração dos modelos de classificação mais precisos baseados em uma amostra de lançamentos financeiros estatisticamente significativo devido a maior quantidade de dados disponível para a geração dos modelos de classificação.

Igualmente nessa direção, novas técnicas poderão ser aplicadas na etapa anterior à geração dos classificadores para que outras informações externas possam ser descobertas e incorporadas como novos atributos para que seja possível, assim, aumentar a acurácia global dos classificadores. Adicionalmente, será desenhado e implementado o processo pelo qual o classificador poderá melhorar seu desempenho com o passar do tempo, levando em conta o preciosas informações de retorno originadas das correções feitas pelos gestores sobre as classificações automáticas.

Por fim, será igualmente importante avaliar se a gestão financeira será menos negligenciada pelos gestores através da automação do processo de categorização de lançamentos financeiros aqui proposto.

Referências

- Al Hasan, Mohammad, et al. (2006) "Link prediction using supervised learning." SDM06: workshop on link analysis, counter-terrorism and security.
- Arlot, Sylvain et al. (2010) A survey of cross-validation procedures for model selection. *Statistics surveys*, v. 4, p. 40-79.
- Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., & Scuse, D. (2010). WEKA manual for version 3-7-3. University of Waikato, New Zealand, 588-595.
- Cortes, Corinna; Vapnik, Vladimir. (1995) Support vector machine. *Machine learning*, v. 20, n. 3, p. 273-297.
- Feldman, R., Sanger, J. (2006): *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press
- Ferreira, Luis Fernando Filardi et al. (2012) Análise quantitativa sobre a mortalidade precoce de micro e pequenas empresas da cidade de São Paulo. *Gestão e Produção*, v. 19, n. 4, p. 811-823.
- Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H., & Trigg, L. (2005). *Weka*. *Data mining and knowledge discovery handbook*, 1305-1314.
- Frank, E., Witten, I. H. (1998): *Generating Accurate Rule Sets Without Global Optimization*. In: *Fifteenth International Conference on Machine Learning*, 144-151.
- Gibbons, J. D., & Chakraborti, S. (2011). *International encyclopedia of statistical science* (pp. 977–979). Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg

- Hunter, John D. Matplotlib (2007): A 2D graphics environment. *Computing In Science & Engineering*, v. 9, n. 3, p. 90-95.
- Imielinski, T., Mannila, H.A. (1996) A database perspective on knowledge discovery. *Commun. ACM* 39,11 (Nov.1996), p. 58-64
- Lakshmi, R.D., Radha, N (2010). Supervised Learning Approach for Spam Classification Analysis using Data Mining Tools (IJCSE) *International Journal on Computer Science and Engineering* Vol. 02, No. 09, 2783-2789
- Leung, K. M (2007). "Naive bayesian classifier." Polytechnic University Department of Computer Science/Finance and Risk Engineering.
- Meyer, D.(2001) "Support vector machines." *R News* 1.3: 23-26.
- Minhas Economias (2017) Disponível em: <http://minhaseconomias.com.br>. Acesso em 09 jul. 2017
- Pang, B.; Lee, L.; Vaithyanathan, S. (2002) Thumbs up?: sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing -Volume 10*. Association for Computational Linguistics. p. 79-86.
- Phan, X.; Nguyen, L.; Horiguchi, S. (2008) Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: *Proceedings of the 17th international conference on World Wide Web*. ACM, p. 91-100.
- Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.
- Roberts, Kirk et al. (2012) EmpaTweet: Annotating and Detecting Emotions on Twitter. In: *LREC*. p. 3806-3813.
- Sajedi, Hedieh; Golazin Zarghami Parast; Fatemeh Akbari. SMS Spam Filtering Using Machine Learning Techniques: A Survey. *Machine Learning Research*. Vol. 1, No. 1, 2016, pp. 1-14. doi: 10.11648/j.ml.20160101.11
- SEBRAE (2004) Fatores Condicionantes e taxa de Mortalidade de Empresas no Brasil 2000-2002, Minas Gerais. SEBRAE (2014) Causa Mortis: o sucesso e o fracasso das empresas nos primeiros cinco anos de vida, São Paulo.
- SEBRAE (2015) Sobrevivência das empresas no Brasil. Coleção estudos e pesquisas. Brasília, Disponível em: <http://goo.gl/baaP7y>. Acesso em 19 mar. 2017
- Souza, W.; Qualharini, E. (2007) O planejamento estratégico nas micro e pequenas empresas. In: *III Workshop Gestão Integrada*. São Paulo.
- Sumner, M., Frank, E., Hall, M. (2005): Speeding up Logistic Model Tree Induction. In: *9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 675-683.
- Vijayarani, S., and M. Divya. (2011)"An efficient algorithm for generating classification rules." *International Journal of Computer Science and Technology* 2.4.
- Wang, S. e D. Manning, C. D. (2012). Baselines and bigrams: simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2 (ACL '12)*, Vol. 2. Association for Computational Linguistics, Stroudsburg, PA, USA, 90-94.
- Wu, X. et al. (2007) Top 10 algorithms in data mining. *Knowl. Inf. Syst.* V. 14,1,p. 1-37.

Evolução de Ontologias em Saúde: Um Estudo de Caso no Domínio da Doença Renal Crônica

Ana Patrícia de Sousa¹, Carlos Alberto de Souza¹, Cecília N. A. P. Gomes²,
Natasha Q. C. Lino¹

¹Programa de Pós-Graduação em Informática – (PPGI), Centro de Informática,
Universidade Federal da Paraíba – (UFPB), João Pessoa – PB, Brasil.

²Programa de Pós-graduação em Modelos de Decisão e Saúde – (PPGMDS),
Universidade Federal da Paraíba – (UFPB), João Pessoa – PB, Brasil.

{anapatrícia, carlossouza}@ppgi.ci.ufpb.br, netapegado@gmail.com,
natasha@ci.ufpb.br

Abstract. *This work proposes a case study that will serve as the basis for the construction of a framework that supports the evolution of ontologies in the health domain. The ontology OntoDecideDRC was used to carry out the case study which resulted in the ONTODRC ontology, name given to the new version of the ontology submitted to the evolution process. The results of this work presented as a contribution a broader and formal vision of the process of evolution of ontologies in the health domain, which will serve as support for the construction of a framework to support this process.*

Resumo. *Este trabalho propõe um estudo de caso que servirá de base para a construção de um framework que ofereça suporte à evolução de ontologias no domínio da saúde. A ontologia OntoDecideDRC foi utilizada para a realização do estudo de caso, o qual obteve como resultado a ontologia ONTODRC, nome dado à nova versão da ontologia submetida ao processo de evolução. Os resultados deste trabalho apresentaram como contribuição uma visão mais ampla e formal do processo de evolução de ontologias no domínio da saúde, o que servirá de apoio à construção de um framework para apoiar esse processo.*

1. Introdução

As ontologias vêm sendo amplamente utilizadas, pois contêm em sua estrutura características como descrição exata do conhecimento, especificação e formalização, que permitem que a informação acessada seja considerada confiável, de modo que a partir dela se gere algum conhecimento útil ao ambiente de sua aplicação, por meio, por exemplo, de inferências e raciocínio dedutível (BREITMAN e LEITE, 2004).

Por ser um processo dinâmico, o desenvolvimento de ontologias deve passar por refinamento de modo que ao longo do tempo possa mudar e se adaptar às necessidades do seu ambiente de aplicação bem como dos seus usuários e evitar a obsolescência do conhecimento nelas contidos. A evolução de ontologias é o processo no qual o sistema de gerenciamento de ontologias de um ambiente permite a realização de mudanças sem comprometer a consistência do conteúdo da ontologia submetida à evolução (STOJANOVIC, 2004).

No mapeamento sistemático realizado por Mota (2014) foram avaliados artigos publicados entre 1986 e 2012 sobre o tema “ontologias e informática médica”. Além de perceber que de 2006 a 2012 houve um enorme crescimento na utilização do tema em trabalhos científicos em relação aos anos anteriores dentro do período da pesquisa, o autor pôde ainda afirmar que 13% deles são relativos à utilização de ontologias para suporte à decisão, podendo então concluir que o suporte à decisão é um uso comum para ontologias.

Assim, tendo em vista o crescimento da utilização de ontologias para suporte à decisão e conseqüentemente a importância do seu processo de evolução, este trabalho apresenta um estudo de caso que servirá de base para a construção de um *framework* para apoiar o processo de evolução de ontologias para suporte à decisão no domínio da saúde. No estudo de caso foi realizada a evolução da OntoDecideDRC, ontologia criada para auxiliar o diagnóstico de pacientes com Doença Renal Crônica (DRC) (TAVARES, 2016).

Este artigo está organizado da seguinte forma: a seção 2 contém a Fundamentação Teórica, que está dividida em 3 subseções: a 2.1 introduz o tema Representação do Conhecimento, detalha o tema Ontologias e discorre sobre Evolução de Ontologias. A subseção 2.2 apresenta uma visão geral sobre a DRC. A 2.3 explana as principais características da OntoDecideDRC. A seção 3 apresenta trabalhos relacionados a Evolução de Ontologias, a mesma está dividida em duas subseções: a 3.1 discorre sobre um trabalho no qual são tratadas abordagens para apoiar a evolução de ontologias no domínio da saúde e a 3.2 na qual é apresentado um trabalho sobre evolução de ontologias em ambientes colaborativos. Na seção 4 foi relatada a metodologia utilizada para a realização do estudo de caso proposto neste artigo. A seção 5 apresenta os resultados do estudo de caso. A 6 traz a conclusão do trabalho bem com suas contribuições. Na seção 7, estão os trabalhos futuros.

2. Fundamentação Teórica

2.1. Representação do Conhecimento, Ontologias e Evolução de Ontologias

Segundo Bueno (2005), representação do conhecimento consiste na especificação dos termos pelos quais o mundo é visto; uma forma de encontrar a abstração considerada correta para que alguém discorra sobre ela e a torne aplicável para a resolução de problemas. A autora completa ainda que diversas teorias da Inteligência Artificial (IA) que representam domínios específicos como medicina, física ou conceitos básicos como tempo e ação são ontologias ou têm uma forte relação com a representação ontológica.

Desde o início dos anos 90, ontologias têm se tornado um tópico especial em pesquisa investigado por várias comunidades de pesquisadores em IA, dentre elas está engenharia do conhecimento, processamento de linguagem natural e representação do conhecimento. Mais recentemente, a noção do termo “ontologia” é também difundido em campos como integração de informação inteligente, informação recuperada na internet e gestão do conhecimento. A principal motivação das ontologias é o fato de que estas permitem o compartilhamento e reuso do conhecimento em computação (STUDER, 1997).

Conforme citado por Guarino (1998) a partir da literatura fundamentada nas bases filosóficas do termo “ontologia”, este vocábulo teve sua origem na filosofia, onde o mesmo é uma descrição sistemática da existência. Em IA, por sua vez, tal expressão é

conceituada por Gruber (1993), como algo que existe e pode ser representado.

Conforme Guarino (1998), o termo “ontologia” deixou de ser apenas uma esfera filosófica do passado e tem ganhado um papel específico no campo da IA, Linguística Computacional e Teoria de Banco de Dados. Para o autor, o termo tem sido reconhecido em áreas de pesquisa tão diversas quanto engenharia do conhecimento, representação do conhecimento, modelagem qualitativa, engenharia da linguagem, projeto de banco de dados, modelagem da informação, integração da informação, análise orientada a objetos, recuperação e extração da informação, organização e gestão do conhecimento, projeto de sistemas baseados em agentes. Guarino (1998) salienta ainda que as áreas onde as ontologias estão sendo aplicadas atualmente são diferentes das anteriormente citadas, elas incluem: integração empresarial, padronização do conhecimento do produto, comércio eletrônico, sistemas de informação geográfica, jurídica e biológica.

Simplificadamente, uma ontologia descreve uma hierarquia de conceitos relacionados pela inclusão de relacionamentos; de um modo mais complexo, axiomas adequados são adicionados de modo ordenado para expressar demais relacionamentos entre conceitos e restringir sua interpretação pretendida (GUARINO, 1998). Para que se possa manter sua consistência, ontologias precisam passar por um processo chamado Evolução de Ontologias.

Evolução de ontologias é a adaptação oportuna de uma ontologia para mudanças que surgirem, e a propagação consistente dessas mudanças para artefatos dependentes. Tendo em vista que mudanças em uma ontologia podem causar inconsistências em suas partes, assim como em objetos dependentes dessa ontologia, a Evolução de Ontologias pode ser considerada como um processo. A evolução de ontologias é muito importante atualmente. A principal razão para isso é o crescimento da sua utilização e o aumento dos custos associados à sua adaptação para refletirem as mudanças requeridas. (STOJANOVIC, 2004).

Considerando que as tarefas realizadas em ontologias como gerenciamento, modificação, evolução e versionamento causam dúvidas relativas às suas definições, Stojanovic (2004) fez distinções entre as mesmas caracterizando cada uma como segue:

- Gerenciamento de Ontologias é todo conjunto de métodos e técnicas necessárias para usar eficientemente múltiplas variantes de ontologias de origens possivelmente diferentes para tarefas distintas;
- A Modificação de Ontologias ocorre quando um sistema de gerenciamento permite mudanças para ontologia que está em uso sem considerar sua consistência;
- A Evolução de Ontologias é realizada quando o sistema de gerenciamento facilita a modificação da ontologia preservando uma consistência;
- O Versionamento de Ontologias acontece quando o sistema de gerenciamento permite manusear as mudanças de uma ontologia por meio da criação e gestão de diferentes versões da mesma.

O sucesso da aplicação de ontologias em ambientes não controlados, descentralizados e distribuídos requer suporte substancial para gerenciamento de mudanças em ontologias bem como para a evolução das mesmas (KLEIN e NOY, 2003). Dada uma ontologia *O* e suas duas versões, *Vnew* e *Vold*, o suporte completo

para o gerenciamento de mudanças no ambiente dessa ontologia inclui apoio à tarefa de transformação de dados, acesso aos dados, atualização da ontologia, raciocínio consistente e verificação e aprovação (KLEIN e NOY, 2003).

Para Stojanovic (2004), o processo de evolução de ontologias é importante porque o desenvolvimento de uma ontologia é dinâmico e a evolução de ontologias possui três desafios: complexidade, pois uma ontologia é um modelo rico e possui inter-relacionamentos que podem aumentar os efeitos das mudanças; dependências, pois uma ontologia pode reusar e estender outras ontologias; e distribuição física, devido ao fato do desenvolvimento de ontologias ser um processo descentralizado e colaborativo.

2.2 Doença Renal Crônica

A Doença Renal Crônica (DRC) é definida como um conjunto de anormalidades na estrutura ou função dos rins e baseia-se em alterações na Taxa de Filtração Glomerular (TFG) e/ou na presença de lesão parenquimatosa que se mantêm por pelo menos três meses. Considera-se portador de DRC, qualquer indivíduo que, independente da causa, apresente $TFG < 60 \text{ mL/min/1,73m}^2$ ou a $TFG > 60 \text{ mL/min/1,73m}^2$ associada a pelo menos um marcador de dano renal parenquimatoso (por exemplo, proteinúria) presente há pelo menos 03 (três) meses (BASTOS E KIRSZTAJN, 2011).

A DRC pode ser classificada em 05 (cinco) estágios de acordo com o grau de redução da TFG e o seu tratamento deve ser intensificado de acordo com a sua evolução. Do primeiro ao terceiro estágio é realizado tratamento conservador, que consiste basicamente no controle dos fatores de risco para a progressão da doença, entre esses fatores estão: hipertensão, paciente idoso, obesidade, histórico familiar, paciente fumante entre outros.

O tratamento pré-dialítico, manutenção do tratamento conservador, é realizado no paciente que esteja no quarto estágio da DRC. A Terapia Renal Substitutiva (TRS), tratamento que consiste na substituição da função renal, pode ser realizada com hemodiálise, diálise peritoneal e transplante renal. Esse tratamento é aplicado a pacientes no último estágio da DRC (NATIONAL KIDNEY FOUNDATION, 2002).

2.3 A Ontologia OntoDecideDRC

A OntoDecideDRC foi formulada para implementar conceitos relacionados aos processos necessários ao diagnóstico, estadiamento e encaminhamento de pacientes com DRC, e para ser construída passou por etapas de conceitualização, formalização e implementação. Segundo Tavares (2016) as três fases mais gerais por que passou a OntoDecideDRC em seu desenvolvimento foram:

- Conceitualização: classificação dos dados e descrição através da elaboração do Léxico Ampliado da Linguagem (LAL), linguagem de representação da engenharia de requisitos utilizada para organizar o vocabulário sobre o domínio em estudo;
- Formalização: mapeamento dos termos conforme as primitivas básicas da construção de ontologias: classes, propriedades, hierarquias, restrições e axiomas.
- Implementação: implementação propriamente dita da ontologia, utilizando a *Ontology Web Language* (OWL), linguagem para suporte à construção de ontologias, que reúne especificações da Web Semântica. Nessa fase, foi utilizada

a ferramenta *Protégé*¹, desenvolvida na Universidade de *Stanford*, cujas principais características são: arquitetura expansível, fácil usabilidade e nível considerável de detalhamento (MATTOS, SIMÕES e FARIAS, 2010).

A *OntoDecideDRC* é constituída por 184 (cento e oitenta e quatro) classes com propriedades de objeto, dados e anotações. A ontologia possui 1.240 (um mil, duzentos e quarenta) axiomas. Desses, 479 (quatrocentos e setenta e nove) são axiomas lógicos. A ferramenta ainda classificou a ontologia como “*SHIF (D)*”, pois a mesma “possui intersecção de conceitos, restrições universais, classes disjuntas, conceitos complexos de negação, propriedades inversas, propriedades funcionais, propriedades de dados, entre outras características” (TAVARES, 2016).

3. Trabalhos Relacionados

3.1 Evolução de Ontologias no Domínio da Saúde

Este trabalho realiza uma investigação sobre abordagens aplicadas à evolução de ontologias no domínio da saúde e apresenta uma proposta de *framework* para realizar um estudo comparativo entre as abordagens. Segundo Gawich et al. (2015) há 5 (cinco) abordagens para evolução de ontologias na área da saúde, são elas: *SemCado*, *DyKOSMap*, *Sub Ontology Evolution in a Distributed Health Care Enterprise*, *Mining Relation Reversals in the Evolution of SNOMED CT² using MapReduce and Semi-Automated Ontology Maneagement (SEAM) System*.

A abordagem *SemCado*, foi proposta por Messaoud, Philippe e Nahla (2015) e propõe uma estratégia para a descoberta causal de algoritmos que detectam o mapeamento entre Redes Bayesianas Causais (CBN, do inglês Causal Bayesian Network) e ontologias. O algoritmo assume o uso de uma única ontologia de domínio. Esse algoritmo representa a extensão do algoritmo My Cado e liga cada nó fornecido pela CBN com cada conceito equivalente fornecido pela ontologia. O objetivo desse algoritmo é descobrir e reusar o conhecimento localizado em CBN para evoluir a ontologia. A maior vantagem dessa abordagem é a reusabilidade de relações causais de redes bayesianas para suporte à evolução de ontologias de domínio (GAWICH et al., 2015).

A abordagem *DyKOSMap*, foi proposta por Reis et al. (2015) e tem por objetivo alterar o mapeamento entre dois conceitos de Sistemas de Organização do Conhecimento (KOS, do inglês Knowledge Organizations Systems) com o intuito de evoluir esse sistema, ou seja, *DyKOSMap* indica a correlação entre a evolução do KOS e o mapeamento da evolução (GAWICH et al., 2015).

Sub Ontology Evolution in a Distributed Health Care Enterprise é uma abordagem que se refere a propagação de mudanças. Proposta por Sari, Rahayu e Bhatt (2013), essa abordagem inclui um grupo de regras para garantir a eficiência da sub ontologia gerada. Seu objetivo é manter a coerência entre as sub ontologias com a

¹ <http://protege.stanford.edu/>

² <http://www.snomed.org/snomed-ct>

ontologia completa. A abordagem de propagação de mudanças depende do uso de logs de mudanças de ontologias do domínio da saúde (GAWICH et al., 2015).

Proposta por Tao et al. (2015), *Mining Relation Reversals in the Evolution of SNOMED CT using MapReduce* é usada para descobrir as relações de reversão hierárquica entre diferentes versões da SNOMED CT de 2009 a 2014 (GAWICH et al., 2015).

Finalmente, *Semi-Automated Ontology Management (SEAM) System* foi proposta por Harris, Livnat e Meystre (2015). Seu objetivo é promover a extração de informações concernentes a doenças específicas. Tais informações podem ser úteis para a construção de uma ontologia (GAWICH et al., 2015).

Em suma, a noção de evolução de ontologias em *Sem Cado Sub Ontology Evolution in a Distributed Health Care Enterprise* é focada na criação de ontologias evoluídas. Outros métodos focam na detecção de inconsistências entre versões de uma ontologia. Apenas SEAM depende de recursos de dados estruturados e não estruturados que forneçam mudanças recentes no domínio médico (GAWICH et al., 2015). O estudo das abordagens descritas nesse trabalho contribuirão com a definição do *framework* ao qual o estudo de caso aqui abordado fornece base, pois através dele é possível identificar características relevantes que devem ser consideradas na definição de uma abordagem para evolução de ontologias.

3.2 Evolução de Ontologias em Ambientes Colaborativos

Conforme Noy et al. (2006), o principal problema do desenvolvimento colaborativo de ontologias é a diversificação de tarefas de manutenção do seu conteúdo, pois essa diversidade requer a utilização de sistemas para suporte diversificados. Em seu trabalho Noy et al. (2006), cita a ferramenta Protegé³ como alternativa viável para a manutenção de ontologias, pois dá suporte à adição de *plugins* específicos para cada atividade que se deseja desempenhar durante esse processo. Ainda assim, a utilização de múltiplos *plugins* é vista pelos autores como um problema, pois dificulta o processo de manutenção. Considerando essa problemática, e com o intuito de saná-la, os autores propuseram um *framework* flexível o suficiente para trabalhar em qualquer ambiente colaborativo, onde o Protégé e seus *plugins* são utilizados para acompanhar ambientes onde não existem relatórios de fases das mudanças realizadas.

No centro desse *framework* está a *Change and Annotation Ontology (CHAO)*: uma ontologia que reúne duas classes principais (*Change* e *Annotation*) e cujas instâncias relatam mudanças específicas e metainformações sobre elas (autor, data e hora, anotações, status de aceitação, entre outras). O gerenciador de mudanças do Protégé é usado para edição de ontologias e os *plugins* criam instâncias CHAO como um produto secundário resultante do processo de edição/manutenção da ontologia (NOY et al., 2006). Os componentes resultantes da modelagem do *framework* proposto por Noy et al., (2006), bem como suas relações, podem ser observados na Figura 1.

³ <http://protege.stanford.edu/>

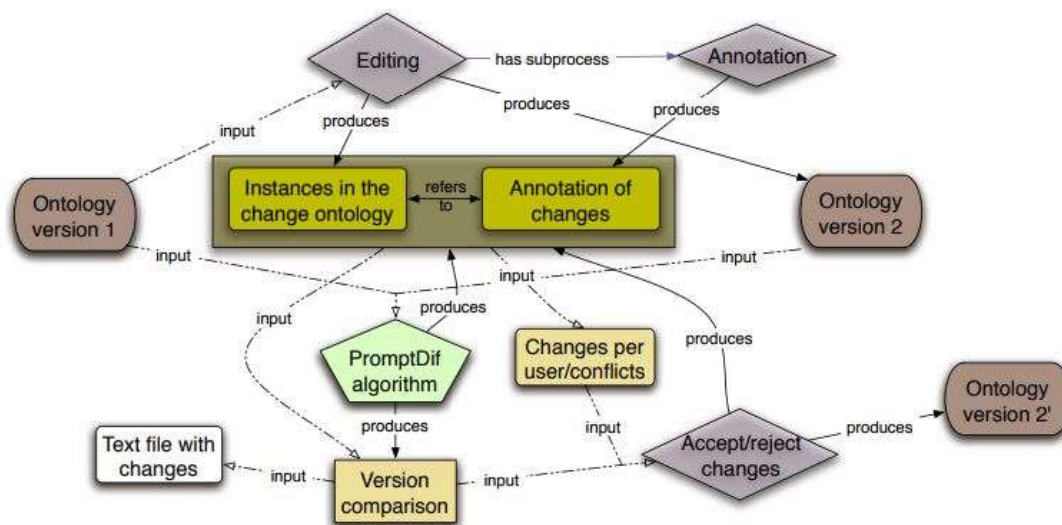


Figura 1. Componentes para do *framework* para evolução de ontologias. Fonte. NOY et al., 2006.

A definição do *framework* cujos componentes estão relacionados na Figura 1 foi baseada em dois *plugins* da ferramenta Protégé: o *Change-management plugin* que fornece acesso a mudanças e histórico e habilita alterações, e o *Prompt plugin* que faz a comparação entre versões e auxilia na tarefa de aceite ou rejeição de mudanças. Por meio desse trabalho foi possível que se identificasse as atividades mais relevantes a serem consideradas no processo de evolução de ontologias em ambientes onde o mesmo necessita ser realizado por vários colaboradores. Desse modo, além de contribuir com a realização do estudo de caso apresentado neste artigo, o *framework* aqui proposto fornecerá apoio à criação de um *framework* para apoiar o processo de evolução de ontologias em saúde, o qual será realizado em trabalhos futuros.

4. Materiais e Métodos

A Figura 2 ilustra a sequência de tarefas executadas para a realização do estudo de caso relatado no presente artigo no qual a ontologia OntoDecideDRC passou por processo de evolução. Essa sequência será descrita nas próximas subseções. O principal objetivo da nova ontologia é reduzir falhas no diagnóstico da DRC, bem como a ausência e ou falta de incorrigibilidade de informações no formulário de encaminhamento do paciente para o nefrologista. A nova versão da ontologia teve seu nome alterado para ONTODRC. A Figura 3 mostra os parâmetros da ontologia antes e depois do processo de evolução.

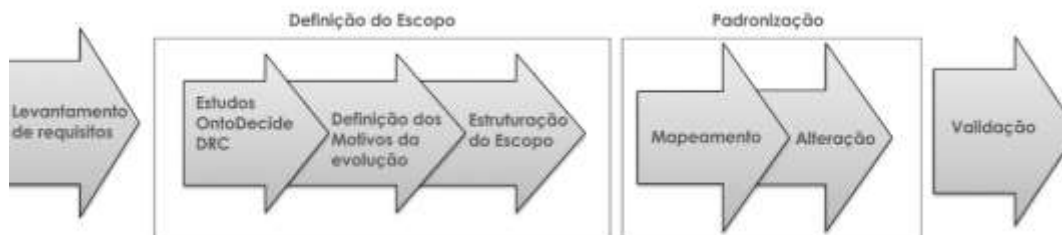


Figura 2. Metodologia de trabalho

4.1 Levantamento de Requisitos

Inicialmente foi feito um levantamento bibliográfico sobre ontologias. Trabalhos clássicos relacionados ao tema como os de Gruber (1993), Studer (1997) e Guarino (1998) foram preferencialmente incluídos nessas pesquisas. Buscas sobre evolução de ontologias foram executadas dando maior ênfase a artigos com aplicações práticas como o trabalho de Noy et. al (2006) e de Klein e Noy (2003).

Nessa fase também foi realizado o estudo das diretrizes clínicas que foram usadas no processo de evolução. Para adaptar a classe da ontologia que faz a classificação da DRC à diretriz internacional, criada e mantida pela *Kidney Disease: Improving Global Outcomes (KDIGO)*⁴, foi incluído na metodologia deste trabalho, o documento de Avaliação e Gerenciamento da DRC (do inglês, *CKD Evaluation and Management*)⁵ fornecido pela KDIGO. O novo foco da ontologia passou a ser o encaminhamento do paciente para a atenção especializada, por isso, além do uso do documento de Diretrizes Clínicas para o cuidado do paciente com Doença Renal Crônica – DRC⁶ no Sistema Único de Saúde (SUS)⁷, que já havia sido utilizado para a criação da versão inicial da OntoDecideDRC, foi utilizado o documento de Protocolos de encaminhamento da Atenção Básica para a Atenção Especializada - Endocrinologia e Nefrologia do Ministério da Saúde⁸.

4.2 Definição do Escopo

A tarefa de definição do escopo da ontologia foi dividida em 3 subtarefas que são: Estudos sobre a OntoDecideDRC, Definição dos Motivos da Evolução, Estruturação do Escopo. Na primeira subtarefa foi realizado um estudo aprofundado do foco da OntoDecideDRC. Logo após, suas classes, relacionamentos e instâncias foram analisadas com a finalidade de entender sua estrutura hierárquica. Na segunda subtarefa, os motivos que justificam a evolução da ontologia foram definidos. O motivo da evolução reside na mudança de foco da OntoDecideDRC, que passou a ser o apoio ao processo de cuidado do paciente antes de ser encaminhado para o Nefrologista. O novo foco da ontologia foi sugerido por um médico especialista em Nefrologia, que se baseou em sua experiência de presenciar a grande ocorrência do diagnóstico impreciso e informações incorretas e inconsistentes, no formulário de encaminhamento do serviço de saúde.

Na subtarefa Estruturação do Escopo, de acordo com o novo foco da ontologia, a estrutura da nova versão foi definida. Nessa fase, os componentes desnecessários e os que deveriam ser adicionados foram definidos com a ajuda do médico especialista. Foram realizadas as operações de mudanças básicas na ontologia, que segundo Klein e Noy (2003), modificam apenas uma característica do modelo de conhecimento OWL⁹, por exemplo, operações de adição ou deleção sobre cada uma dessas características.

⁴ <http://kdigo.org>

⁵ <http://kdigo.org/guidelines/ckd-evaluation-and-management/>

⁶ http://bvsmis.saude.gov.br/bvs/publicacoes/diretrizes_clinicas_cuidado_paciente_renal.pdf

⁷ <http://portalsaude.saude.gov.br/index.php/cidadao/entenda-o-sus>

⁸ http://dab.saude.gov.br/portaldab/biblioteca.php?conteudo=publicacoes/protocolos_atencao_basica_atencao_especializada

⁹ Ver <https://www.w3.org/TR/owl-features/>

Adicionalmente também foram realizadas mudanças complexas, que são compostas por múltiplas operações básicas, ou aquelas que contêm algum conhecimento adicional sobre a mudança, como por exemplo, uma operação de mudança sobre relações de superclasse.

4.3 Padronização

Após a estruturação do escopo foi realizada a padronização da nomenclatura da ontologia submetida ao processo de evolução. O padrão escolhido para essa tarefa foi o SNOMED CT, que substituiu o padrão usado até então, que era baseado nas diretrizes para o cuidado ao paciente com DRC no SUS. SNOMED CT é uma terminologia clínica que pode ser usada para representar clinicamente informações relevantes de forma consistente, confiável e compreensiva como uma parte integrante da produção de informações eletrônicas de saúde (SNOMED CT, 2017).

A tarefa de padronização foi dividida em duas etapas: Mapeamento, no qual termos correspondentes aos da ontologia foram pesquisados no *browser* SNOMED CT¹⁰ e Alteração, processo no qual os termos da ontologia foram substituídos, respeitando suas definições e estrutura hierárquica.

4.4 Validação

A última tarefa do estudo de caso consistiu na validação, onde o nefrologista avaliou, com a ajuda da equipe responsável pela evolução, a estrutura e os conceitos presentes na ontologia resultante de modo a verificar sua consistência e confiabilidade ao ser aplicada para o suporte à decisão clínica.

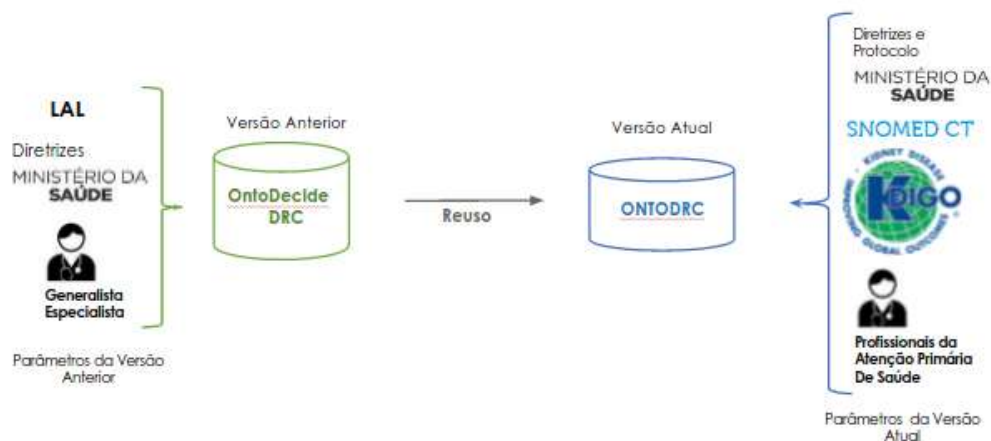


Figura 3. Parâmetros da ontologia, antes e depois do processo de evolução.

5. Resultados

A seguir são apresentadas as principais classes da ontologia OntoDecideDRC que sofreram mudanças básicas e/ou complexas:

- A classe “Classificação” foi renomeada para “Classificação/Estadiamento”. A operação de “Mover” foi realizada sobre a classe “Relação Albuminúria/Creatinúria” que passou a ser subclasse de

¹⁰ <http://browser.ihtsdotools.org/>

“Classificação/Estadiamento” à qual foi adicionada outra subclasse chamada “Estadiamento da Função Renal”. A Figura 4A apresenta a estrutura hierárquica da classe Classificação/Estadiamento. Na Figura 4B é possível observar os sinônimos criados para a classe “Doença Renal Crônica Estágio 3”.

- A subclasse “Outros Motivos de Encaminhamento” foi adicionada à classe “Encaminhamento” e apresenta como subclasses, doenças que também podem ser encaminhadas para o nefrologista e que se apresentam como fatores de risco da DRC.
- A nova classe “Profissionais da Atenção básica de Saúde” pode ser vista na Figura 5A.
- A nova classe “Outros motivos de encaminhamento” pode ser observada na Figura 5B.

Nenhuma propriedade da ontologia sofreu alteração. A validação da ontologia resultante também foi realizada pelo nefrologista que considerou confiáveis os termos nela usados, tendo em vista que foram baseados em diretrizes reconhecidas nacional e internacionalmente.

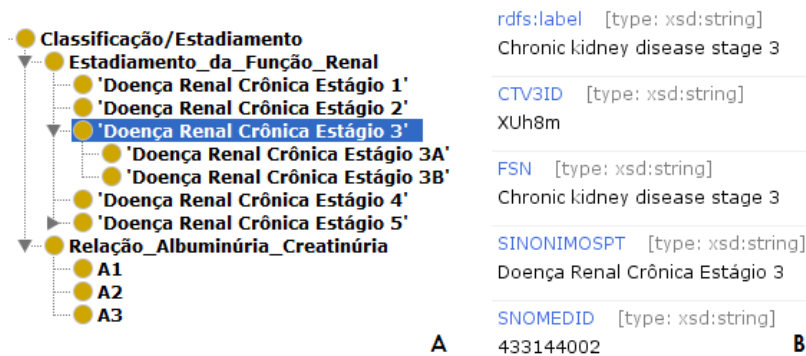


Figura 4. A. Classe “Classificação/Estadiamento. B. Sinônimos “Doença Renal Crônica Estágio 3”.

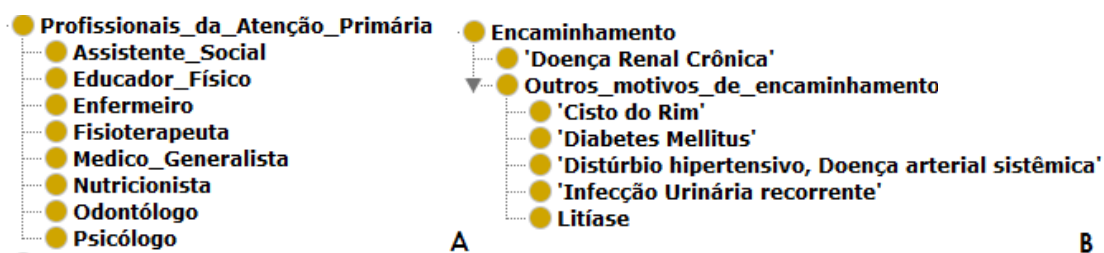


Figura 5. A. Classe “Profissionais da Atenção Primária”. B. Classe “Outros motivos de encaminhamento”.

Na Figura 6 podem ser observadas as métricas das duas versões da ontologia. Devido à mudança de foco e conseqüentemente a redução do escopo da nova versão, a quantidade de classes e axiomas da nova versão são menores.

Ontology metrics:	OntoDecideDRC	ONTODRC
Metrics		
Axiom	1150	768
Logical axiom count	389	280
Declaration axioms count	252	174
Class count	184	102
Object property count	26	26
Data property count	4	4
Individual count	15	15
Annotation Property count	29	31
DL expressivity	SHIF(D)	SHIF(D)

Figura 6. Métricas das versões da ontologia.

6. Conclusão

Após a evolução, a nova ontologia (ONTODRC) passou a refletir conceitos mais específicos em relação à sua versão anterior. Em vez de abordar todo o processo de tratamento da DRC, a ontologia foca nas fases iniciais da doença, com o intuito de minimizar falhas como diagnóstico impreciso e informações incorretas e inconsistentes no formulário de encaminhamento, e como consequência reduzir o retrabalho no tratamento da doença. O estudo de caso apresentou como contribuição uma visão mais ampla da importância da evolução do conhecimento contido em ontologias particularmente quando esse conhecimento é relativo ao suporte à decisão em saúde. A utilização da SNOMED CT permitiu uma visão mais abrangente da terminologia, bem como das suas vantagens na padronização de termos clínicos. A presença do médico especialista na realização do estudo de caso permitiu a definição precisa do motivo da evolução e foi imprescindível para a validação da ontologia resultante.

Portanto, pode-se concluir que os métodos utilizados para a realização do estudo de caso relatado neste artigo ofereceram subsídios comprovadamente eficientes para dar aporte ao processo de evolução de ontologias utilizadas para suporte à decisão no domínio da saúde. Como esperado, o estudo também proporcionou uma base teórica e prática para a formulação das tarefas que irão compor um *framework* para apoiar esse processo.

7. Trabalhos Futuros

Como trabalhos futuros pretende-se definir formalmente as tarefas do *framework* citado, cujo principal objetivo será subsidiar o processo de evolução de ontologias para suporte à decisão clínica.

Tendo em vista que o processo de padronização dos termos da ontologia em processo de evolução é uma tarefa trabalhosa e pode aumentar os custos (com recursos humanos, financeiros e de tempo) do processo de evolução, propõe-se como trabalhos futuros a automatização da padronização desses termos utilizando a base de dados SNOMED CT, e recursos de desenvolvimento de *software*.

Referências

- BASTOS, M. G., e Kirsztajn, G. M. (2011). Doença renal crônica: importância do diagnóstico precoce, encaminhamento imediato e abordagem interdisciplinar estruturada para melhora do desfecho em pacientes ainda não submetidos à diálise. *J bras nefrol*, 93-108.
- BUENO, Tânia C. D'Agostini. (2005) Uma metodologia de representação do conhecimento para construção de ontologias em sistemas baseados em conhecimento. 2005. 174 f. Tese (Doutorado em Engenharia) - Universidade Federal de Santa Catarina, Florianópolis.
- BREITMAN, K. K. J. C. LEITE, S. P. (2004) Ontologias: Como e porque criá-las. Pontifícia Universidade Católica. Rio de Janeiro.
- GAWITH, M.; ALFONSE, M.; MOSTAFA, A.; SALEM, A. M. (2015) "Ontology Evolution Approaches in Medical Domain". *International Journal of Biology and Biomedical Engineering*.
- GRUBER, T. R. (1993) *A translation approach to portable ontology specifications*. Knowledge Acquisition, vol. 5, pp. 199–220.
- GUARINO, N. *Formal Ontology and Information Systems* (1998). Proceedings of FOIS'98, Trento, Italy, 6-8. Amsterdam, IOS Press, pp. 3-15.
- KLEIN, M. e NOY, N. F. (2003) *A component-based framework for ontology evolution*. Workshop on Ontologies and Distributed Systems at IJCAI-03, Acapulco, Mexico.
- MOTA, M. R. A. ; LINO, N. C. Q. ; TAVARES, E. A. ; CAMPOS, S. P. R. . (2014) Um Mapeamento Sistemático sobre o Uso de Ontologias em Informática Médica. In: CBIS 2014 - XIV Congresso Brasileiro de Informática em Saúde, 2014, Santos. Anais CBIS 2014 - XIV Congresso Brasileiro de Informática em Saúde.
- NATIONAL KIDNEY FOUNDATION. (2002). *Clinical practice guidelines for chronic kidney disease: evaluation, classification and stratification*. National Kidney Foundation.
- NOY, N. F.; CHUGH, A.; LIU, W. e MUSEN, M. A. (2006) *A Framework for Ontology Evolution in Collaborative Environments*. *International Semantic Web Conference (ISWC)*.
- SNOMED CT. (2017) *What is SNOMED CT?*. Disponível em: <http://www.snomed.org/snomed-ct/what-is-snomed-ct>. Acesso em: 21 de Junho de 2017.
- STOJANOVIC, L. (2004) *Methods and tools for Ontology Evolution*. 2004. 249 f. Tese (Doutorado) – *Universitaet Fridericana zu Karlsruhe*, Karlsruhe.
- STUDER, R.; BENJAMINS, V. R.; FENSEL, D. (1998) *Knowledge Engineering: Principles and methods*. *Data & Knowledge Engineering*, 25 (1-2), 161-197.
- TAVARES. E. A., Gomes. C. N. A. P., Lino. N. C. Q. . (2016). Suporte à Decisão Clínica no Domínio da Doença Renal Crônica. XV Congresso Brasileiro de Informática em Saúde - Goiânia - Brasil. PP. 839-847

PACTL-SYM: um planejador baseado em Verificação Simbólica de Modelos

Viviane Bonadia dos Santos¹, Leliane Nunes de Barros¹

¹Instituto de Matemática e Estatística – Universidade de São Paulo (USP)
Rua do Matão, 1010 - CEP 05508-090 - São Paulo - SP

{vbonadia, leliane}@ime.usp.br

Resumo. Planejamento clássico em inteligência artificial, cujo objetivo é encontrar um plano de ações que alcance um estado meta a partir de um estado inicial, faz suposição de ambientes completamente observáveis e ações com efeitos determinísticos. Uma extensão de planejamento clássico, chamada de FOND (Fully-Observable Non-Deterministic Problem), considera ações com efeitos não determinísticos e a geração de políticas que eventualmente terminam em becos-sem-saídas. Soluções conhecidas para esse problema são baseadas em técnicas de verificação de modelos sendo que a maioria delas utiliza a lógica temporal de tempo ramificado CTL. Para contornar as limitações da lógica CTL, que não leva em consideração as ações, foi proposta uma nova lógica, a lógica α -CTL, e um planejador para problemas FOND, chamado PACTL. Neste trabalho apresentamos a versão simbólica desse planejador, que representa estados e ações como fórmulas lógicas, utiliza técnicas de verificação simbólica de modelos e diagramas de decisão binária (BDDS).

1. Introdução

Em inteligência artificial, planejamento consiste no processo de seleção de ações que ao serem executadas garantem atingir um conjunto de metas preestabelecidas \mathcal{G} a partir de um estado inicial s_0 [Nau et al. 2004]. No planejamento clássico são feitas diversas suposições restritivas sobre o ambiente que o agente atua, entre elas: o ambiente é completamente observável, discreto, finito e determinístico, isto é, não existe incerteza sob os efeitos das ações executadas pelo agente. Uma classe de planejadores que lidam com problemas que possuem menos restrições considera ações com efeitos não determinísticos, chamados de FOND (Fully-Observable Non-Deterministic Problem).

A Figura 1 apresenta um exemplo esquemático de domínios de planejamento em que os nós representam estados do ambiente e as arestas representam as ações que ao serem executadas em um estado levam o agente a outro estado. A Figura 1(a) ilustra um domínio de planejamento clássico e 1(b) um domínio de planejamento não determinístico. Note que, ao executar a ação c no estado s_2 no domínio não determinístico, o agente pode ir para o estado s_3 ou para o estado s_4 . A solução para um problema de planejamento clássico é uma sequência totalmente ordenada de ações; enquanto a solução para um problema FOND é um plano ramificado, isto é, uma política que associa uma ação a cada estado.

Um exemplo de planejador considerado estado-da-arte para problemas FOND é o planejador PRP (Planner for Relevant Policies) [Muisse et al. 2012]. Apesar do PRP ser eficiente, ele não garante devolver a solução ótima, isto é, uma política que contém o

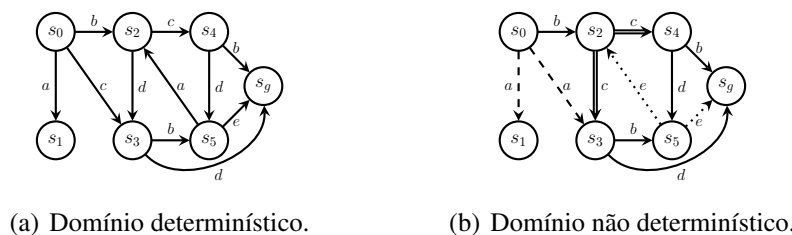


Figura 1. Domínio de planejamento em que s_0 é o estado inicial e $s_g \in \mathcal{G}$. Arestas rotuladas pela mesma ação representam efeitos não determinísticos.

caminho de menor distância para um estado meta. O PRP utiliza o planejador FD (*Fast Downward*) [Helmert 2006] (um planejador baseado em heurísticas considerado estado-da-arte em planejamento clássico) para resolver uma versão determinística do problema FOND e, em seguida, regride as ações do plano encontrado para obter uma solução mais geral para o problema FOND original. Em geral, o PRP devolve um plano subótimo.

Outro planejador que se destaca na área de planejamento não determinístico é o planejador MBP (*Model Based Planner*) [Bertoli et al. 2001], que utiliza técnicas de verificação simbólica de modelos. A ideia principal desta abordagem é resolver problemas de planejamento através de um modelo formal, dando garantia sobre a validade do plano gerado. Além disso, o MBP permite especificar metas complexas que não podem ser resolvidas pelo planejador PRP. Ao longo dos anos, diferentes métodos de verificação de modelos foram propostos sendo que, muitos deles, representavam explicitamente o espaço de estados através de tabelas. Contudo, o número de estados do modelo pode crescer exponencialmente, fazendo com que o tamanho das tabelas seja um fator limitante em sistemas realísticos. A verificação simbólica de modelos é uma abordagem usada para controlar o problema de explosão de estados em que: conjuntos de estados são representados por fórmulas lógicas e a exploração do espaço de estados é feita através de operações lógicas. Tais operações podem ser realizadas de forma eficiente utilizando diagramas de decisão binária (BDDs- *Binary decision diagram*) [Bryant 1992].

Diversos planejadores que utilizam técnicas de verificação simbólica se destacaram na Competição Internacional de Planejamento (IPC)¹ de 2014. Quatro dentre os cinco melhores colocados na categoria de planejadores clássicos ótimos (*Sequential Optimal track*) eram planejadores que utilizavam BDDs [Edelkamp et al. 2015]. O planejador GAMER [Kissmann and Edelkamp 2008] que competiu na categoria de planejadores FOND, no IPC de 2008, também utiliza BDDs.

Neste trabalho apresentamos um planejador baseado em verificação simbólica de modelos e na lógica α -CTL. A semântica α -CTL permite que o planejador resolva problemas da classe FOND raciocinando formalmente sobre as ações. Diferentemente do MBP que utiliza mecanismos extra-lógicos para extrair políticas, α -CTL permite um planejamento que faça verificação enquanto sintetiza um plano [Pereira and de Barros 2008]. Este artigo está organizado da seguinte forma: na Seção 2 apresentamos os conceitos de planejamento como verificação de modelos; na Seção 3 mostramos como representar estados e ações através de fórmulas lógicas bem como as operações fundamentais de um planejador baseado em verificação simbólica de modelos; na Seção 4 descrevemos o planejador PACTL-SYM; na Seção 5 mostramos um exemplo de planejamento no Robô de

¹<http://www.icaps-conference.org/index.php/Main/Competitions>

Carga; e, finalmente, na Seção 6 apresentamos as conclusões e trabalhos futuros.

2. Fundamentos

2.1. O problema de planejamento FOND

Um domínio de planejamento é uma tupla $\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T} \rangle$, sendo que: \mathcal{S} é um conjunto finito de estados do ambiente; \mathcal{A} é um conjunto finito de ações que o agente pode executar; e $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow 2^{\mathcal{S}}$ é uma função de transição de estados que leva o agente de um estado $s \in \mathcal{S}$ para um possível conjunto de estados $X \subseteq \mathcal{S}$ através de uma ação $a \in \mathcal{A}$. Um problema de planejamento, por sua vez, é definido por uma tupla $\mathcal{P} = \langle \mathcal{D}, s_0, \mathcal{G} \rangle$ em que: \mathcal{D} é o domínio de planejamento; $s_0 \in \mathcal{S}$ é o estado inicial do ambiente; e $\mathcal{G} \subseteq \mathcal{S}$ é o conjunto de estados meta.

Em decorrência da existência de ações com efeitos não determinísticos, a solução para um problema FOND pode resultar em múltiplos caminhos. Neste contexto, a solução para um problema de planejamento não determinístico é uma *política* π , isto é, um mapeamento de estados em ações. A política determina a ação mais apropriada para ser executada em cada estado e pode ser classificada em três tipos: *fraca*, *forte* e *forte-cíclica*. A política fraca é uma política que eventualmente pode alcançar um estado meta. Um exemplo de política fraca para o domínio da Figura 1(b) é o conjunto $\pi = \{(s_0, a), (s_3, d)\}$; a política forte é uma política que, a despeito do não determinismo, garante alcançar um estado meta. Por exemplo, na Figura 1(b) a política $\pi = \{(s_0, b), (s_2, c), (s_3, d), (s_4, b)\}$ é uma política forte; e a política forte-cíclica é uma política que sempre alcança um estado meta pressupondo que sua execução eventualmente conseguirá sair de todos os ciclos existentes. Na Figura 1(b), $\pi = \{(s_0, b), (s_2, c), (s_3, b), (s_4, d), (s_5, e)\}$ é um exemplo de política forte-cíclica.

2.2. Verificação de modelos usando a lógica CTL

Verificação de modelos é uma técnica de verificação formal que consiste na exploração de um sistema de transição finito (modelo) a fim de verificar a validade de uma dada propriedade. Dado um modelo \mathcal{M} , um estado s e uma propriedade φ , um verificador de modelos verifica $(\mathcal{M}, s) \models \varphi$, devolvendo sucesso quando \mathcal{M} satisfaz φ a partir do estado s ou um contra-exemplo, caso contrário [Clarke et al. 1999]. Uma abordagem comum em verificação de modelos é a formalização do modelo \mathcal{M} através de uma estrutura de Kripke. Seja \mathbb{P} um conjunto finito não vazio de proposições atômicas, uma estrutura de Kripke sobre \mathbb{P} é uma tupla $\mathcal{K} = \langle \mathcal{S}, T, \mathcal{L} \rangle$ em que: \mathcal{S} é um conjunto finito não vazio de estados; $T : \mathcal{S} \rightarrow 2^{\mathcal{S}}$ é uma função de transição de estados (sem considerar as ações); e $\mathcal{L} : \mathcal{S} \rightarrow 2^{\mathbb{P}}$ é uma função de rotulação de estados. A propriedade a ser verificada, por sua vez, é formalizada através de uma fórmula da lógica CTL.

A lógica CTL (*Computation Tree Logic*) é uma lógica temporal de tempo ramificado que permite especificar propriedades quantificadas sobre caminhos em uma árvore de computação. Existem quatro operadores nesta lógica: \circ (próximo estado), \diamond (finalmente, em um estado futuro), \square (globalmente, em todos os estados) e \sqcup (uma fórmula é válida até que outra seja verdade). Todo operador temporal é precedido de um quantificador existencial \exists (existe algum caminho) ou universal \forall (em todos os caminhos).

Definição 1 (Sintaxe CTL) O conjunto de fórmulas CTL é definido indutivamente como:
 $\varphi ::= \neg\varphi_1 \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \exists \circ \varphi_1 \mid \forall \circ \varphi_1 \mid \exists \square \varphi_1 \mid \forall \square \varphi_1 \mid \exists \diamond \varphi_1 \mid \forall \diamond \varphi_1 \mid \exists(\varphi_1 \sqcup \varphi_2) \mid \forall(\varphi_1 \sqcup \varphi_2)$.

A semântica de uma fórmula CTL é dada em termos de uma estrutura de Kripke. De acordo com a semântica CTL dizemos, por exemplo, que $(\mathcal{M}, s) \models p$ é verdadeira se o estado s é rotulado com a proposição p ; e $(\mathcal{M}, s) \models \forall \diamond p$ é verdadeira se todos os caminhos na estrutura de Kripke, a partir de s , finalmente alcançam um estado que satisfaz p . A Figura 2 apresenta caminhos computacionais em modelos que satisfazem fórmulas da lógica CTL (por simplicidade omitimos o operador \circ). Por exemplo, a Figura 2(e) ilustra que todos os caminhos de \mathcal{M} a partir de s , finalmente alcançam um estado que satisfaz φ .

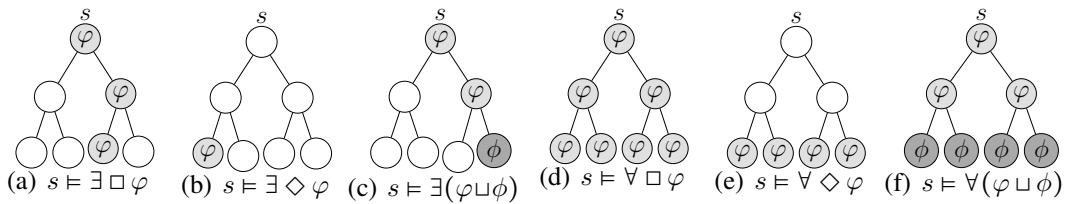


Figura 2. Semântica dos operadores na árvore computacional CTL. Nós em cinza claro representam estados do modelo que satisfazem φ e nós em cinza escuro estados que satisfazem ϕ .

Os algoritmos de verificação de modelos CTL baseiam-se na implementação de operações de pré-imagem. Dado um conjunto de estados $X \subseteq \mathcal{S}$, a operação de pré-imagem computa um conjunto de estados Y que alcançam X . Existem dois tipos de pré-imagem: a *pré-imagem fraca* (que computa os estados Y que podem alcançar X , contudo, pode existir uma ramificação a partir de algum estado de Y que não alcança X) e a *pré-imagem forte* (que computa os estados que garantidamente alcançam X).

$$\begin{aligned} \text{PREIMAGEMFRACA}(X) &= \{s \in \mathcal{S} \mid T(s) \cap X \neq \emptyset\} \\ \text{PREIMAGEMFORTE}(X) &= \{s \in \mathcal{S} \mid \emptyset \neq T(s) \text{ e } T(s) \subseteq X\} \end{aligned}$$

Figura 3. Operações de pré-imagem de verificação de modelos em CTL.

Note que para verificar uma fórmula temporal do tipo $\exists \diamond \varphi$, usamos a pré-imagem fraca e para a fórmula do tipo $\forall \diamond \varphi$, usamos a pré-imagem forte. O Algoritmo 1 [Giunchiglia and Traverso 2000] apresenta uma função de um verificador de modelos que verifica se uma estrutura de Kripke satisfaz uma fórmula CTL do tipo $\exists \diamond \varphi$ a partir de um estado inicial s . Por falta de espaço, os algoritmos que tratam os demais operadores CTL foram omitidos.

Algoritmo 1: VERIFICAEF(\mathcal{K}, s, φ)

```

1  estadosAtuais ← ∅
2  proximosEstados ← ESTADOS( $\varphi, \mathcal{K}$ ) // estados da estrutura de Kripke que satisfazem  $\varphi$ 
3  enquanto proximosEstados ≠ estadosAtuais faça
4      se  $s \in proximosEstados$  então
5          retorna Verdadeiro
6      fim
7      estadosAtuais ← proximosEstados
8      proximosEstados ← proximosEstados ∪ PREIMAGEMFRACA(proximosEstados)
9  fim
10 retorna Falso
    
```

2.3. Planejamento baseado em verificação de modelos com α -CTL

No planejamento baseado em verificação de modelos, o modelo descreve o domínio de planejamento $\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T} \rangle$. Note que, ao invés de uma estrutura de Kripke, em planejamento temos um modelo cujas transições são rotuladas pelas ações. Enquanto um verificador de modelos verifica a validade de determinada propriedade em um modelo \mathcal{M} , um planejador procura um submodelo de \mathcal{M} (política) que satisfaça a meta a partir do estado inicial. Assim, dado um problema de planejamento $\mathcal{P} = \langle \mathcal{D}, s_0, \mathcal{G} \rangle$, \mathcal{D} é um modelo de transição de estados rotulado pelas ações, e \mathcal{G} é a propriedade φ que desejamos verificar no modelo a partir de s_0 .

Neste contexto, os planejadores baseados em verificação de modelos que utilizam CTL necessitam de alguns mecanismos extra-lógicos para auxiliar na escolha de ações (dado que a estrutura de Kripke não contém ações). A lógica α -CTL é uma extensão de CTL que permite verificar a validade de uma propriedade e sintetizar políticas sem a necessidade de mecanismos extra-lógicos, uma vez que sua semântica leva em consideração as diferentes transições causadas pelos diferentes tipos de ações [Pereira 2007]. A lógica α -CTL é composta por quatro operadores temporais: \odot (*sucessor imediato*), \square (*invariantemente*), \diamond (*finalmente*) e \sqcup (*até que*) que devem, obrigatoriamente, ser precedidos por um quantificador de caminho: \forall ou \exists (o “ponto” nos operadores referem-se às ações).

Definição 2 (Sintaxe α -CTL) *Seja $p \in \mathbb{P}$ uma proposição atômica. A sintaxe de α -CTL é definida indutivamente como:*

$$\varphi ::= \top \mid p \mid \neg p \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid \exists \odot \varphi \mid \forall \odot \varphi \mid \exists \square \varphi \mid \forall \square \varphi \mid \exists(\varphi_1 \sqcup \varphi_2) \mid \forall(\varphi_1 \sqcup \varphi_2).$$

A semântica de uma fórmula α -CTL é dada em termos de um sistema de transição de estados rotulados por ações, isto é, uma estrutura de Kripke em que as transições são rotuladas pelas ações. Por exemplo: (1) em α -CTL um modelo \mathcal{M} satisfaz $\exists \odot p$ a partir de s se existe uma ação $a \in \mathcal{A}$ que, quando executada em s , possivelmente alcança um sucessor imediato s' que satisfaz p (isto é, algum efeito não determinístico de a leva ao estado s'); (2) em α -CTL o modelo \mathcal{M} satisfaz $\forall \odot p$ a partir de s se existe uma ação $a \in \mathcal{A}$ que quando executada em s , necessariamente alcança um sucessor imediato s' que satisfaz p (isto é, todos os efeitos não determinísticos de a levam ao estado s').

Seja \mathcal{M} um modelo de transição de estados rotulado por ações e $s \in \mathcal{S}$ o estado inicial. A Figura 4 ilustra a semântica dos operadores \square , \diamond e \sqcup precedidos pelo quantificador de caminho \forall . Note que as transições são rotuladas pelas ações a, b, c e d . Na figura 4(b), por exemplo, $s \models \forall \diamond \varphi$ expressa que para todos os estados futuros, *alguma* ação leva a estados que finalmente satisfazem φ . Observe a diferença entre a semântica dos operadores \diamond e \diamond quando precedidos pelo operador \forall (figuras 2(e) e 4(b) respectivamente). Em CTL a fórmula é satisfeita a partir de um estado s se todos os estados futuros, sem diferenciar as transições, finalmente alcançam um estado que satisfaz φ , enquanto em α -CTL a fórmula é satisfeita se existe uma ação em que todos os efeitos não determinísticos finalmente alcançam um estado que satisfaz φ . Uma descrição mais detalhada da semântica α -CTL pode ser encontrada em [Pereira 2007].

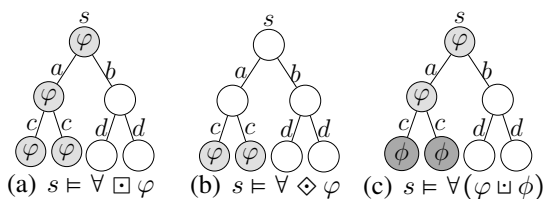


Figura 4. Semântica dos operadores \square , \diamond e \sqcup na árvore computacional α -CTL.

Com base na semântica da lógica α -CTL foi proposto o planejador PACTL (*Planejador Alpha CTL*) [Pereira 2007], que baseia-se na implementação de operações de pré-imagem. O cálculo da pré-imagem fraca de um conjunto de estados $X \subseteq \mathcal{S}$ (usado na síntese de política fraca) computa um conjunto de pares estado-ação que possivelmente alcançam X e o cálculo da pré-imagem forte (usado na síntese de política forte) computa um conjunto de pares estado-ação que garantidamente (a despeito do não determinismo) alcançam X . A Figura 5 apresenta as definições de pré-imagem baseadas em α -CTL.

$$\begin{aligned} \text{PRÉIMAGEMFRACA}_{actl}(X) &= \{(s, a) : s \in \mathcal{S}, a \in \mathcal{A} \text{ e } \mathcal{T}(s, a) \cap X \neq \emptyset\} \\ \text{PRÉIMAGEMFORTE}_{actl}(X) &= \{(s, a) : s \in \mathcal{S}, a \in \mathcal{A} \text{ e } \emptyset \neq \mathcal{T}(s, a) \subseteq X\} \end{aligned}$$

Figura 5. Operações de pré-imagem α -CTL.

O Algoritmo 2 apresenta uma função do planejador PACTL que computa um submodelo $M_\pi \subseteq \mathcal{D}$ a partir do qual é possível extrair uma política fraca. Esta função recebe como parâmetro um problema FOND $\mathcal{P} = \langle \mathcal{D}, s_0, \varphi \rangle$, sendo φ a meta de planejamento \mathcal{G} descrita em α -CTL. Esta função primeiramente determina o conjunto de estados que satisfazem φ (Linha 1). Em seguida, iterativamente, este conjunto é expandido através de operações de pré-imagem fraca até alcançar um ponto-fixo ($M_\pi \neq M'_\pi$), a partir do qual podemos garantir que todos os pares (s, a) do submodelo computado finalmente alcançam um estado $s \models \varphi$ através de algum efeito não determinístico de a . Em [Pereira 2007] são descritos algoritmos de planejamento para cada um dos operadores da lógica α -CTL.

Uma meta de planejamento pode ser especificada em α -CTL de três formas: $\exists \diamond \varphi$ se uma solução fraca é desejada; $\forall \diamond \varphi$ se uma solução forte é desejada; ou $\forall \square \exists \diamond \varphi$ se uma solução forte-cíclica é desejada. Em [Pereira and de Barros 2008, Pereira 2007], foram tratadas fórmulas α -CTL que expressam metas de alcançabilidade estendidas, dadas por um par (φ_1, φ_2) em que φ_1 especifica uma condição que deve ser preservada ao longo da execução da política e φ_2 uma propriedade que deve ser satisfeita no final da execução da política. Por exemplo, $\forall(\varphi_1 \sqcup \varphi_2)$ expressa uma meta de alcançabilidade estendida quando uma solução forte é desejada.

Algoritmo 2: MODELOEF(\mathcal{P}) // síntese de políticas fracas

```

1  $M_\pi \leftarrow \{s \mid s \in \mathcal{S} \text{ e } s \models \varphi\}$  // conjunto de estados que satisfazem  $\varphi$ 
2  $M'_\pi \leftarrow \emptyset$ 
3 enquanto  $M_\pi \neq M'_\pi$  faça
4      $M'_\pi \leftarrow M_\pi$ 
5      $C \leftarrow \{s \mid \exists a \in \mathcal{A} \text{ e } (s, a) \in M_\pi\}$  // conjunto de estados pertencentes a  $M_\pi$ 
6      $I \leftarrow \text{PRÉIMAGEMFRACA}_{actl}(C)$ 
7      $M_\pi \leftarrow M_\pi \cup I$ 
8 fim
9 retorna  $M_\pi$ 

```

3. Planejamento baseado em verificação simbólica de modelos com α -CTL

O planejamento baseado em verificação simbólica de modelos faz buscas num espaço de conjuntos de estados descritos por fórmulas lógicas. Esses conjuntos de estados, em geral, são extremamente grandes. A representação deles usando diagramas de decisão binária resulta em uma representação compacta e que permite operações computacionalmente eficientes.

3.1. Diagramas de Decisão Binária

Diagramas de Decisão Binária (BDD - *Binary Decision Diagram*) são grafos direcionados acíclicos (DAG) que representam funções $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Um BDD pode ser descrito como o resultado de simplificações em uma árvore de decisão. Uma árvore de decisão, por sua vez, é uma estrutura que possui nós de decisão rotulados com variáveis booleanas e nós terminais rotulados com 1 (verdadeiro) ou 0 (falso). Cada nó v de decisão possui dois nós sucessores: $l(v)$ que representa a atribuição do valor falso à variável v , e $h(v)$ que representa a atribuição do valor verdadeiro à v [Huth and Ryan 2004].

Embora árvores de decisão sejam muito úteis para representar funções booleanas, em geral, elas podem possuir muita redundância. Quando tais redundâncias são eliminadas, dá-se o nome para estrutura resultante dessas eliminações de *diagrama de decisão binária*, uma forma mais compacta para representação de fórmulas booleanas. O termo BDD, em geral, refere-se a diagrama de decisão binária reduzido ordenado (ROBDD - *Reduced Ordered Binary Decision Diagram*). Além da representação compacta, para toda expressão booleana existe um único ROBDD que o representa. Por simplicidade, quando a notação BDD é utilizada, faz-se referência a ROBDD. A Figura 6 ilustra um exemplo de uma árvore de decisão e de um BDD para fórmula $\varphi = (x_1 \vee x_2) \wedge (x_2 \vee x_3)$.

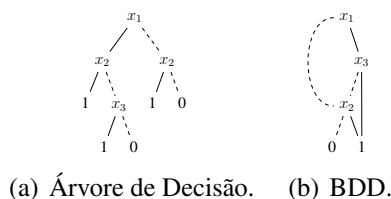


Figura 6. Árvore de decisão para a fórmula $\varphi = (x_1 \vee x_2) \wedge (x_2 \vee x_3)$ e BDD que representa φ de forma compacta. A aresta tracejada representa $x_i = \text{Falso}$ e a aresta sólida representa $x_i = \text{Verdadeiro}$.

3.2. Regressão simbólica

Em planejamento baseado em verificação simbólica de modelos, os conjuntos de estados e ações do problema são representados através de fórmulas booleanas quantificadas (QBF - *Quantified Boolean Formulas*) [Büning and Bubeck 2009], e a busca através do espaço de estados é realizada fazendo transformações lógicas sobre as fórmulas que representam os conjuntos de estados. Seja $\mathcal{D} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T} \rangle$ um domínio de planejamento sobre um conjunto de proposições \mathbb{P} . Um estado $s \in \mathcal{S}$ é representado simbolicamente como:

$$\xi(s) = \bigwedge_{p \in \mathbb{P}(s)} p, \quad (1)$$

e um conjunto de estados $X \subseteq \mathcal{S}$ é representado como:

$$\xi(X) = \bigvee_{s \in X} \xi(s). \quad (2)$$

Neste trabalho, ao invés de representarmos de forma simbólica o modelo de transição de estados rotulado pelas ações e proposições (como é feito pela maioria dos planejadores baseados em verificação de modelos), representamos o modelo através da descrição de ações STRIPS. Assim, podemos construir o espaço de estados apenas com a especificação de s_0 , \mathcal{G} e um conjunto de ações STRIPS. Uma ação STRIPS [Fikes and Nilsson 1971] $a \in \mathcal{A}$ é definida em termos de suas precondições e efeitos. As precondições da ação a , denotadas por $Precond(a)$, representam as proposições que devem ser verdadeiras no estado em que a será executada. Os efeitos, por sua vez, são subdivididos em dois conjuntos: efeitos positivos, denotados por $Ef^+(a)$, que representam as proposições que se

tornam verdadeiras após a execução de a ; e efeitos negativos, denotados por $Ef^-(a)$, que representam as proposições que se tornam falsas após a execução de a . No caso das ações não determinísticas, os efeitos são representados por um conjunto de efeitos não determinísticos: $Efeitos(a) = \{e_1, \dots, e_n\}$ onde cada efeito e_i possui seu próprio conjunto de efeitos positivos (Ef^+) e negativos (Ef^-).

No planejamento baseado em verificação simbólica de modelos é necessário representar as ações através de fórmulas QBF. Assim, uma ação não determinística a será representada por uma tupla $\langle \xi(Precond(a)); \xi(Efeitos(a)); Mudanças(a) \rangle$ tal que: $\xi(Precond(a))$ é a fórmula que representa $Precond(a)$ e $\xi(Efeitos(a))$ a fórmula que representa $Efeitos(a)$, respectivamente definidos pelas fórmulas 3 e 4 sendo p, t e r átomos da linguagem; e $Mudanças$ é um conjunto de proposições que estão em $Efeitos(a)$.

$$\xi(Precond(a)) = \left(\bigwedge_{p \in Precond(a)} p \right). \quad (3)$$

$$\xi(Efeitos(a)) = \left(\bigvee_{e_i \in Efeitos(a)} \left(\bigwedge_{t \in Ef^+(a, e_i)} t \right) \wedge \left(\bigwedge_{r \in Ef^-(a, e_i)} \neg r \right) \right). \quad (4)$$

Para raciocinar sobre este modelo utilizamos operações de regressão pelas ações. A operação de regressão por uma ação $a \in \mathcal{A}$ é similar à operação de pré-imagem α -CTL (Figura 5), porém é feita sobre as fórmulas que descrevem as ações, isto é, em termos de condições e efeitos. Com base na semântica de QBF, temos uma série de equivalências das fórmulas booleanas, dentre elas, duas que valem destacar e que serão utilizadas para computar a regressão são: $\forall x\varphi \equiv \varphi[0/x] \wedge \varphi[1/x]$; e $\exists x\varphi \equiv \varphi[0/x] \vee \varphi[1/x]$. Em tais equivalências, x é uma variável proposicional que ocorre na fórmula φ .

O cálculo da regressão fraca ($REGFRACA(X, a)$) computa a fórmula que representa o conjunto de estados que podem alcançar o subconjunto $X \subseteq \mathcal{S}$ através da ação a . O cálculo da regressão forte ($REGFORTE(X, a)$) computa a fórmula que representa o conjunto de estados que garantidamente alcançam o subconjunto $X \subseteq \mathcal{S}$ através da ação a [Menezes et al. 2014]. É possível provar que as regressões fraca e forte (Figura 7) equivalem a um passo da pré-imagem fraca e forte, respectivamente, (Figura 5) segundo a ação a [Rintanen 2008]. Note que, nas operações de regressão fraca e forte os quantificadores \exists e \forall são da lógica QBF.

$\begin{aligned} REGFRACA(X, a) &= \xi(Precond(a)) \wedge \exists Mudanças(a).(\xi(X) \wedge \xi(Efeitos(a))) \\ REGFORTE(X, a) &= \xi(Precond(a)) \wedge \forall Mudanças(a).(\xi(Efeitos(a)) \rightarrow \xi(X)) \end{aligned}$

Figura 7. Operações simbólicas de regressão na lógica QBF.

4. O planejador PACTL-SYM

O PACTL-SYM (*Planejador Alpha-CTL Simbólico*) é um planejador baseado em verificação simbólica de modelos. Dado um problema de planejamento FOND $\mathcal{P} = \langle \mathcal{D}, s_0, \phi \rangle$ definido sobre um conjunto de proposições \mathbb{P} , em que \mathcal{D} é dado pelas ações representadas conforme as Equações 3 e 4 e ϕ é a meta de planejamento especificada em α -CTL, o planejador, num primeiro momento, computa um submodelo $M_\pi \subseteq \mathcal{D}$. Se $s_0 \subseteq M_\pi$ então uma política π , extraída de M_π , é devolvida, caso contrário o planejador devolve falha.

O planejador PACTL-SYM raciocina sobre conjuntos de estados (representados conforme a Equação 2) e explora o espaço de estados utilizando operações de regressão fraca e forte (Figura 7). Computacionalmente os estados e ações são representados utilizando BDDs, sendo que cada proposição do problema é uma variável do BDD.

O Algoritmo 3 apresenta a função do planejador que computa um submodelo de uma solução fraca ($\exists \diamond \varphi$). A cada iteração i do algoritmo é computada a regressão por todas as ações $a \in \mathcal{A}$ e o resultado armazenado em uma camada i de estados e ações (par de BDD representando os estados computados pelas regressões e conjunto de ações que geraram estes estados). A primeira camada do submodelo armazena um BDD que representa os estados que satisfazem \mathcal{G} , a partir desta camada o algoritmo expande o submodelo, através do cálculo de regressão, de forma que cada nova camada possui um conjunto de estados que pode alcançar o conjunto de estados da camada anterior. O algoritmo para ao encontrar uma camada que contém o estado inicial ou quando alcança um ponto fixo. A divisão do submodelo em camadas permite uma representação bastante compacta dos conjuntos de estados e faz com que as operações da busca sejam mais eficientes. Para computar a regressão, o algoritmo chama a função auxiliar REGRESSAOFraca (Algoritmo 4). Dado um conjunto de estados $\xi(X)$ esta função computa um conjunto de estados $\xi(Y)$ que podem alcançar $\xi(X)$. Para isso, esta função aplica a operação de regressão fraca, definida na Figura 7, para cada uma das ações do problema. Além dos estados que podem alcançar $\xi(X)$, esta função retorna um conjunto de ações que ao serem aplicadas em estados pertencentes a Y levam a estados pertencentes a X . A função VERIFICAINCONSISTENCIA (Linha 4) verifica se o estado gerado é inconsistente, isto é, se proposições mutuamente exclusivas são verdadeiras no estado gerado. Caso sejam, a função retorna \perp , senão o próprio estado é retornado. Assumimos que o conjunto de proposições mutuamente exclusivas é dado para o algoritmo.

Algoritmo 3: MODELOEF(\mathcal{P}) // síntese de políticas fracas

```

1  $\xi(M_\pi)^{BDD} \leftarrow \bigvee_{s \models \varphi} \xi(s)$  // conjunto de estados que satisfazem  $\varphi$ 
2  $\xi(M'_\pi)^{BDD} \leftarrow \perp$ ;  $i = 0$ 
3  $camadas_i.inserer(\xi(M_\pi)^{BDD}, \emptyset)$ 
4 enquanto  $\xi(s_0) \notin \xi(M_\pi)^{BDD}$  ou  $\xi(M_\pi)^{BDD} \neq \xi(M'_\pi)^{BDD}$  faça
5      $\xi(M'_\pi)^{BDD} \leftarrow \xi(M_\pi)^{BDD}$ 
6      $(\xi(Y)^{BDD}, acoes) \leftarrow \text{PRÉIMAGEMFRACA}(\mathcal{P}, \xi(M_\pi)^{BDD})$ 
7      $i = i + 1$ ;  $camadas_i.inserer(\xi(Y)^{BDD}, acoes)$ 
8      $\xi(M_\pi)^{BDD} \leftarrow \xi(Y)^{BDD}$ 
9 fim
10 retorna  $camadas$ 
```

Algoritmo 4: REGRESSAOFraca($\mathcal{P}, \xi(X)^{BDD}$) // regressão simbólica

```

1  $\xi(Y)^{BDD} \leftarrow \perp$ ;  $acoess \leftarrow []$ 
2 para  $a \in A$  faça
3      $\xi(I)^{BDD} \leftarrow \xi(\text{Precond}(a)) \wedge \exists \text{Mudanças}(a).(\xi(X) \wedge \xi(\text{Efeitos}(a)))$ 
4      $\xi(I)^{BDD} \leftarrow \text{VERIFICAINCONSISTENCIA}(\xi(I)^{BDD})$ 
5     se  $\xi(I)^{BDD} \neq \perp$  então
6          $\xi(Y)^{BDD} \leftarrow \xi(Y)^{BDD} \vee \xi(I)^{BDD}$ 
7          $acoess.inserer(a)$ 
8     fim
9 fim
10 retorna  $(\xi(Y)^{BDD}, a)$ 
```

A função de síntese de política forte ($\forall \diamond \varphi$) é similar à função de síntese de política fraca, porém, ao invés de chamar a função auxiliar REGRESSAOFraca, é feita uma chamada para a função auxiliar REGRESSAOFORTE que substitui a operação de regressão

fraca (Linha 3) pela operação de regressão forte conforme apresentado na Figura 7.

Se o estado inicial está contido na última camada, então existe uma solução para o problema. Para extrair uma política realizamos uma busca para frente a partir do estado inicial utilizando operações de progressão. Dado um estado s a operação de progressão, denotada por $Progr(s, a, e_i)$ (Fórmula 5), computa o estado que pode ser alcançado, segundo o efeito não determinístico e_i da ação a , quando esta é executada em s .

$$Progr(s, a, e_i) = \exists Mudanças(a, e_i). (\xi(s) \wedge \xi(Precond(a))) \wedge \xi(Efeitos(a, e_i)). \quad (5)$$

Dado um submodelo com n camadas, se s_0 está contido no conjunto de estados da última camada (camada n), o algoritmo computa os estados S que s_0 pode alcançar considerando apenas as ações da camada n , sendo que S é um ou mais estados contidos no conjunto de estados armazenados na camada $n - 1$. Em seguida, o algoritmo computa os estados S' que podem ser alcançados a partir de S considerando apenas as ações armazenadas na camada $n - 1$ e assim sucessivamente até que o estado meta seja alcançado. A cada passo de progressão, para cada estado gerado é associado uma ação. O conjunto de pares estado-ação computados na busca progressiva representa a política extraída a partir do submodelo computado pelo Algoritmo 3. A Figura 12(a) ilustra um exemplo esquemático do funcionamento do algoritmo de extração da política.

Neste artigo nos concentramos em descrever o funcionamento dos algoritmos que computam soluções fracas e fortes. Resumidamente, para computar a política forte-cíclica, combinamos as operações de regressão fraca e forte de forma que, num primeiro momento o algoritmo computa, através da regressão fraca, um conjunto de estados que podem alcançar a meta. Em seguida, este conjunto é contraído, através da regressão forte, de forma que pares de estado-ação que podem levar para algum estado que não está presente em nenhuma camada sejam eliminados. Desta forma, todos os estados presentes no submodelo resultante eventualmente alcançam a meta.

5. Planejamento para o problema do Robô de Carga

Para exemplificar o funcionamento do planejador PACTL-SYM vamos considerar o domínio do Robô de Carga em que um robô deve transportar n caixas de uma localização para outra. Como exemplo, vamos considerar um cenário em que temos 2 salas (sala A e sala B) e 1 caixa que deve ser transportada de uma sala B para sala A (Figura 8).

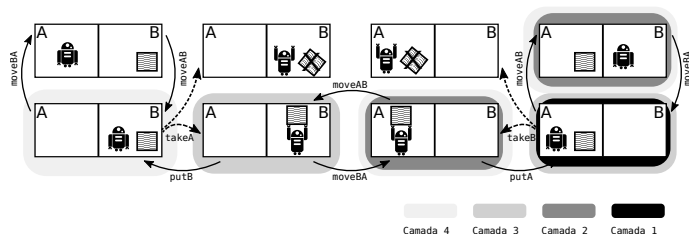


Figura 8. Domínio do robô de carga

Este problema pode ser representado pelo seguinte conjunto de proposições \mathbb{P} : $boxAtA$ (representa que a caixa está no chão da sala A); $boxAtB$ (a caixa está no chão da sala B); $boxOnr$ (a caixa está nos braços do robô); $boxOk$ (a caixa não está quebrada); $rAtA$ (o robô está na sala A); $rAtB$ (o robô está na sala B) e $rFree$ (o robô não está carregando nenhuma caixa). As proposições mutuamente exclusivas deste problema são: $\{rAtA, rAtB\}$ pois o robô não pode estar em duas salas ao mesmo tempo e

$\{boxAtA, boxAtB, boxOnr\}$ pois, ou a caixa está no chão de uma das salas, ou nos braços do robô. O robô pode executar seis ações, sendo elas: $moveAB$, move da sala A para sala B; $moveBA$, move da sala B para sala A; $putA$, descarrega a caixa na sala A; $putB$, descarrega a caixa na sala B; $takeA$, carrega a caixa na sala A; e $takeB$, carrega a caixa na sala B. A Figura 9 descreve as ações $moveBA$, $putA$ e $takeB$ de maneira simbólica. Note que, sempre que o robô tenta carregar uma caixa ele poderá derrubá-la no chão e quebrá-la ($\neg boxOk$) e assim o robô não poderá mais transportá-la para outra localização.

$moveBA$:	$\langle \xi(Precond(moveBA)) = rAtB; \xi(Efeitos(moveBA)) = rAtA \wedge \neg rAtB$ Mudanças = $\{rAtA, rAtB\}$
$putA$:	$\langle \xi(Precond(putA)) = boxOnr \wedge rAtA; \xi(Efeitos(putA)) = boxAtA \wedge rFree \wedge \neg boxOnr$ Mudanças = $\{boxAtA, rFree, boxOnr\}$
$takeB$:	$\langle \xi(Precond(takeB)) = boxAtB \wedge rAtB \wedge rFree \wedge boxOk;$ $\xi(Efeitos(takeB)) = (boxOnr \wedge \neg boxAtB \wedge \neg rFree) \vee \neg boxOk;$ Mudanças = $\{boxOnr, boxAtB, rFree, boxOk\}$

Figura 9. Ações simbólicas do domínio do robô de carga.

Seja o estado inicial $s_0 = (boxAtB \wedge boxOk \wedge rAtB \wedge rFree \wedge \neg boxAtA \wedge \neg boxOnr \wedge \neg rAtA)$ e $\mathcal{G} = \exists \diamond (rAtA \wedge boxAtA \wedge boxOk)$, a Figura 10 mostra o resultado da regressão fraca a partir de \mathcal{G} considerando a ação $moveBA$ e a Figura 11 apresenta o resultado das camadas construídas de acordo com o Algoritmo 3.

$$\begin{aligned}
 REGRFRACA(\xi(\mathcal{G}), moveBA) &= \xi(Precond(moveBA)) \wedge \exists Mudanças(moveBA).(\mathcal{G} \wedge \xi(Efeitos(moveBA))) \\
 &= rAtB \wedge \exists rAtA, rAtB.((boxAtA \wedge rAtA \wedge boxOk) \wedge (rAtA \wedge \neg rAtB)) \\
 &= rAtB \wedge \exists rAtA, rAtB.(boxAtA \wedge rAtA \wedge boxOk \wedge \neg rAtB) = rAtB \wedge \exists rAtB.(boxAtA \wedge boxOk \wedge \neg rAtB) \\
 &= rAtB \wedge boxAtA \wedge boxOk
 \end{aligned}$$

Figura 10. Regressão fraca a partir de \mathcal{G} segundo a ação $moveBA$.

A extração da política parte do estado inicial e realiza uma busca para frente através do cálculo de progressão (Fórmula 5), considerando apenas as ações da camada 4. A Figura 12 apresenta um exemplo de progressão a partir de s_0 segundo a ação $takeB$ e o efeito não determinístico $e_i = (boxOnr \wedge \neg boxAtB \wedge \neg rFree)$. O par estado-ação resultante dessa operação é: $(\xi(s_0), takeB)$. Quando $takeB$ é executada em s_0 o robô pode ir para o estado $\xi(s) = rAtB \wedge \neg boxAtA \wedge \neg rAtA \wedge boxOnr \wedge \neg boxAtB \wedge \neg rFree$.

Camada	Conjunto de estados $\xi(X)$	Ações
1	$(boxAtA \wedge rAtA \wedge boxOk)$	\emptyset
2	$(boxAtA \wedge rAtB \wedge boxOk) \vee (boxOnr \wedge rAtA \wedge boxOk)$	$moveAB, putA$
3	$(boxAtA \wedge rAtA \wedge boxOk \wedge rFree) \vee (boxOnr \wedge rAtB \wedge boxOk)$	$moveAB, takeA, moveBA$
4	$(boxAtA \wedge rAtB \wedge boxOk) \vee (boxOnr \wedge rAtA \wedge boxOk) \vee (boxAtB \wedge rAtB \wedge boxOk)$	$moveBA, putA, moveAB, takeB$

Figura 11. Submodelo construído pelo Algoritmo 3 para o Robô de Carga.

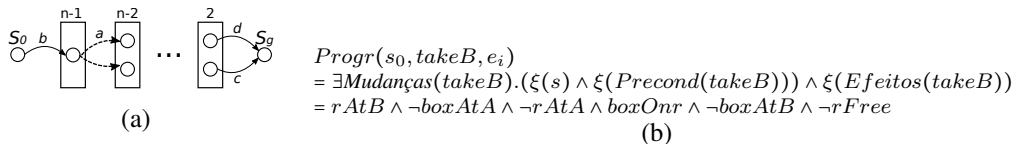


Figura 12. a) Exemplo esquemático de extração da política a partir do submodelo computado. b) Estado alcançado a partir de s_0 segundo a ação $takeB$ considerando o efeito $e_i = (boxOnr \wedge \neg boxAtB \wedge \neg rFree)$.

6. Conclusão e trabalhos futuros

Um dos grandes desafios de planejamento automatizado é construir algoritmos eficientes que devolvam soluções ótimas. Nos últimos anos diversas técnicas para resolver problemas de planejamento da classe FOND foram propostas. Dentre elas, uma técnica que se

destaca é a baseada em verificação de modelos. A maioria dos planejadores dentro desta abordagem são baseados em CTL. Porém, ela não é adequada para formalizar algoritmos de síntese de política uma vez que sua semântica não considera os diferentes tipos de transições, causadas pelas diferentes ações do domínio. Neste trabalho apresentamos um planejador para problemas FOND, baseado em verificação simbólica de modelos e na lógica α -CTL, que permite construir algoritmos de síntese de maneira formal. A abordagem simbólica permite representar grandes conjuntos de estados e raciocinar sobre eles de forma eficiente com o uso de BDDs. Implementamos os algoritmos apresentados utilizando C++ e a biblioteca para manipulação de BDDs CUDD². Trabalhos futuros envolvem executar testes em outros domínios e problemas e um estudo comparativo do nosso planejador com outros planejadores FOND, entre eles os MBP e o PRP.

Agradecimentos Este trabalho recebeu suporte do CNPq, CAPES-PROEX e FAPESP (projeto 2015/01587-0).

Referências

- Bertoli, P., Cimatti, A., Pistore, M., Roveri, M., e Traverso, P. (2001). MBP: a model based planner. *Workshop on Planning under Uncertainty and Incomplete Information*.
- Bryant, R. E. (1992). Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv.*, 24(3):293–318.
- Büning, H. K. e Bubeck, U. (2009). Theory of Quantified Boolean Formulas. *Handbook of Satisfiability*, páginas 735–760.
- Clarke, E. M., Grumberg, O., e Peled, D. (1999). *Model checking*. MIT Press.
- Edelkamp, S., Kissmann, P., e Torralba, A. (2015). BDDs strike back (in AI planning).
- Fikes, R. E. e Nilsson, N. J. (1971). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *IJCAI'71*, páginas 608–620.
- Giunchiglia, F. e Traverso, P. (2000). Planning as model checking. *Proceedings of the 5th European Conference on Planning: Recent Advances in AI Planning*.
- Helmert, M. (2006). The fast downward planning system. *J. Artif. Int. Res.*
- Huth, M. e Ryan, M. (2004). *Logic in Computer Science: Modelling and Reasoning About Systems*. Cambridge University Press.
- Kissmann, P. e Edelkamp, S. (2008). Gamer: Fully-observable non-deterministic planning via pddl-translation into a game.
- Menezes, M. V., Barro, L. N., e Pereira, S. L. (2014). Symbolic regression for non-deterministic actions. *Learning and Nonlinear Models*, 12:98–114.
- Muise, C. J., McIlraith, S. A., e Beck, J. C. (2012). Improved non-deterministic planning by exploiting state relevance. *ICAPS*.
- Nau, D., Ghallab, M., e Traverso, P. (2004). *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Pereira, S. L. (2007). *Planejamento sob incerteza para metas de alcançabilidade estendidas*. Tese de doutorado, Universidade de São Paulo.
- Pereira, S. L. e de Barros, L. N. (2008). A logic-based agent that plans for extended reachability goals. *Autonomous Agents and Multi-Agent Systems*, 16(3):327–344.
- Rintanen, J. (2008). Regression for classical and nondeterministic planning. *European Conference on Artificial Intelligence*, páginas 568–572.

²<http://vlsi.colorado.edu/fabio/CUDD/html/index.html>

Treinamento Otimizado de Redes Neurais Artificiais Para a Classificação de Movimentos Musculares

Fábio K.H. de Barros¹, Aron A. Martins L.², Fabio Augusto Guidotti dos Santos²,
Victor Hideki Yoshizumi², Rafael Martineli de Araujo², Maria Eugênia Dajer²,
Danilo Hernane Spatti²

¹Departamento de Computação – Universidade Tecnológica Federal do Paraná (UTFPR)
Cornélio Procópio – PR – Brasil

²Departamento de Elétrica – Universidade Tecnológica Federal do Paraná (UTFPR)
Cornélio Procópio – PR – Brasil

{fabiobarros, aron, fsantos.1995, yoshizumi, rafaelaraujo}@alunos.utfpr.edu.br
medajer@utfpr.edu.br, danilospatti@utfpr.edu.br

Abstract. *The use of Intelligent systems (ANNs, Fuzzy, Adaptive Systems) emerged in the last few years in several areas, one of them being bioengineering, where these tools are essential for pattern recognition in areas such as voice, muscle, brain signals and so on. In these areas, an example of a very powerful tool is the use of Artificial Neural Networks. A big problem of Artificial Neural Networks is the empirical search for the best topology. Thus, this article proposes the implementation of a Hybrid Intelligent System with educational purposes, capable of recognizing patterns of electromyographic signals and using Artificial Neural Networks alongside Genetic Algorithms.*

Resumo. *A aplicação de sistemas inteligentes (RNAs, Fuzzy, Sistemas Adaptativos) emergiu nos últimos anos em diversas áreas, sendo uma delas a bioengenharia, onde é essencial a utilização destes para reconhecimento de padrões, como por exemplo em sinais de voz, musculares, cerebrais, entre outros. Para esta aplicação, um exemplo de ferramenta muito poderosa são as Redes Neurais Artificiais. Um grande problema das Redes Neurais Artificiais é a busca empírica pela melhor topologia. Sendo assim, este trabalho propõe-se a realizar a implementação de um Sistema Inteligente Híbrido com fins educacionais, a partir de Redes Neurais Artificiais e Algoritmos Genéticos, capaz de reconhecer padrões de sinais eletromiográficos.*

1. Introdução

Atualmente, as Redes Neurais Artificiais (RNAs) emergem como uma poderosa ferramenta para solucionar uma ampla gama de problemas, podendo ser consideradas até mesmo como essenciais em aplicações como o reconhecimento de padrões [Ramchoun et al. 2016] [Ivanov and Gavrilauá 2014].

No entanto, ainda existe muito espaço para evolução no estudo das RNAs. Um exemplo claro disto é o ajuste dos parâmetros relacionados à sua topologia, tais como o número de camadas neurais escondidas, quantidade de neurônios nestas camadas e taxa de aprendizado, sendo que esse pode ser considerado um dos principais desafios encontrados

no desenvolvimento de RNAs e também parte essencial de qualquer trabalho que faça o uso das mesmas, pois a escolha correta para esses valores pode afetar diretamente o resultado final [Ramchoun et al. 2016] [Ivanov and Gavriláú 2014] [Kamiński 2016].

Estes ajustes influenciam diretamente na acurácia da rede, pois se tais parâmetros forem ajustados de forma equivocada (com a utilização de mais neurônios que o necessário, por exemplo), pode-se até desnecessariamente aumentar a carga computacional envolvida na operação [Çam et al. 2015].

A forma mais comum usada no ajuste destes parâmetros é aquela feita manualmente pelo projetista da rede, de forma empírica, basicamente utilizando-se de tentativa e erro, o que pode dificultar o trabalho e consumir muito tempo de quem está desenvolvendo a RNA [Çam et al. 2015]. Desta maneira, como diversos parâmetros podem ser modificados nestas redes, e isto configura-se como sendo uma tarefa de grande dificuldade, este problema pode até mesmo ser considerado como NP-difícil. Devido a isso, surge a necessidade de se auxiliar o desenvolvedor da rede, e para isso, existem no meio acadêmico trabalhos que fazem o uso de uma solução automatizada na geração de topologias de RNAs, como no trabalho de [Fermino et al. 2016].

Entretanto, a otimização também tem ganhado espaço e tem sido vista como uma abordagem promissora neste campo de estudo [da Silva Severo 2010].

No campo da otimização computacional, uma das abordagens mais utilizadas é a dos Algoritmos Genéticos (AGs). Estes configuram-se como uma poderosa ferramenta de busca aleatória direcionada inspirada na teoria da evolução das espécies, formulada por Charles Darwin, onde sobrevivem os indivíduos mais aptos para um determinado ambiente. Isto faz deles a abordagem mais adequada para a otimização da topologia das RNAs pois o cérebro humano, uma das inspirações para as RNAs, nada mais é do que o resultado de um processo de evolução biológico [Correa et al. 2011] [da Silva Severo 2010].

Para realizar esta otimização, é necessário utilizar estas duas técnicas de Sistemas Inteligentes (SIs) em conjunto. Quando dois SIs diferentes são usados em conjunto, tem-se a formação de um Sistema Inteligente Híbrido (SIH), cujo principal objetivo é unir os pontos fortes de cada ferramenta.

Como já dito anteriormente, as RNAs tem a capacidade de solucionar problemas nas mais diversas áreas do conhecimento. Em [Ivanov and Gavriláú 2014] por exemplo, os autores utilizam RNAs em conjunto com um AG para otimização de um parâmetro relativo à RNA, da mesma maneira que este trabalho. Entretanto, o parâmetro otimizado é a função de ativação para cada camada neural. O principal objetivo deste sistema é ser um estimador otimizado de picos de carga elétrica em um sistema de distribuição de energia.

Já em [Ramchoun et al. 2016], os autores realizaram a implementação de um sistema híbrido que foi usado na classificação de padrões da já consolidada base de dados Iris, utilizando o número de camadas neurais escondidas e a quantidade de neurônios nas mesmas na otimização da estrutura da RNA, de forma similar ao presente artigo.

Em [Belov and Zolotov 2015] foi desenvolvido um otimizador de parâmetros de RNAs aplicado à área de Sistemas de Controle de Objetos Eletromecânicos, onde, de forma análoga a este trabalho, o AG foi usado para fazer o ajuste de parâmetros como

número de neurônios nas camadas escondidas e taxas de aprendizado.

Por fim, em [Çam et al. 2015] os autores realizaram a otimização de parâmetros como: número de neurônios nas camadas neurais ocultas e taxa de aprendizado, assim como neste trabalho, mas empregando diversas técnicas de otimização diferentes, tais como: AGs, Colônias Artificiais de Abelhas (CAA), e Enxame de Partículas (EP).

Tendo em vista o exposto, a proposta deste trabalho foi utilizar-se de dois SIs amplamente utilizados para desenvolver um SIH com fins educacionais, a ser empregado nos algoritmos elaborados em atividades práticas de RNAs. O objetivo do mesmo é atuar no ajuste otimizado dos parâmetros da RNA e também o treinamento da mesma de forma ótima, para que o resultado final seja a melhor rede treinada para um determinado conjunto de dados de entrada.

No caso do presente trabalho, foi usado um banco de dados de sinais mioelétricos, os quais utilizam-se da eletromiografia (EMG) para realizar a leitura de pulsos elétricos musculares sobre a pele. Estes sinais são empregados principalmente no controle de próteses de membros superiores. Para isso, é necessária a identificação do movimento pretendido, o que pode ser obtido a partir do reconhecimento de padrões, e como já foi citado anteriormente no texto, as RNAs são uma poderosa ferramenta para esta classe de problemas [Herle et al. 2010] [Al-Timemy et al. 2015] [Oskoei and Hu 2007].

Por essa razão, para aumentar a acurácia na identificação destes movimentos, este trabalho propõe uma ferramenta para auxiliar na otimização da topologia das redes e treiná-las visando obter a melhor RNA para a classificação dos movimentos.

2. Metodologia

2.1. Dados para estudo de caso

O banco de dados que será utilizado no estudo de caso fora extraído do trabalho disponível em [Khushaba et al. 2016], e consiste em classes de movimento: (C1) - Mão fechada, (C2) - Mão aberta, (C3) - Extensão do Pulso (C4) - Flexão do Pulso, (C5) - Desvio Ulnar do Pulso, (C6) - Desvio Radial do Pulso. Cada uma destas classes foram subdivididas em 9 subclasses que definiam 3 orientações diferentes do pulso em 3 diferentes intensidades: fraca, média e forte. Para cada uma destas subclasses intrínsecas às classes, foram realizados 3 testes para minimizar possíveis erros, gerando assim $3 \times 3 \times 3 \times 6 = 162$ dados amostrados com taxa de 4000 Hz. Para este trabalho será utilizada apenas a classe C1-Mão Fechada.

2.2. Preparação dos dados

A análise de sinais representa uma das mais importantes ferramentas no processamento digital de sinais. Essa análise pode ser realizada por meio das transformadas de Fourier, porém, uma das grandes deficiências da análise de Fourier surge do fato de que ela não apresenta um caráter local, mas sim um “comportamento global médio”, além de não fornecer uma análise temporal, apenas frequencial [Oliveira 2007a].

A transformada *wavelet* (WT) é uma transformada linear que pode ser utilizada na análise de sinais não estacionários para extrair informações das variações em frequência desses sinais e para detectar suas estruturas temporalmente e/ou espacialmente localizadas [Castilho et al. 2012].

Deste modo, a WT compõe uma ferramenta moderna, com uma larga gama de aplicações que permite unificar um vasto número de técnicas de processamento e análise, como a análise de sinais médicos, análise de imagens, codificação em sub-bandas, análise multirresolução para visão artificial em computadores e modelagem de sistemas variantes no espaço-tempo [Oliveira 2007a].

Utilizando o conceito de análise em multirresolução, é possível construir famílias de *wavelets* discretas e desenvolver algoritmos rápidos para o cálculo da transformada. A transformada *wavelet* contínua é calculada fazendo translações e escalonamentos contínuos de uma função sobre um sinal, o que demanda muito tempo e recursos computacionais, além de gerar um alto nível de redundância. Com o intuito de superar este obstáculo, foram introduzidas as *wavelets* discretas, pois, elas não são escalonadas nem transladadas continuamente, mas sim em intervalos discretos [Oliveira 2007a].

Com base na ferramenta de análise multi-resolução que estabelece os princípios básicos para a determinação de uma *wavelet*, pode-se criar funções *wavelet* a partir de certas condições desejadas de regularidade local. Isso permitiu que diversos pesquisadores desenvolvessem estudos e criassem sua própria família de *wavelets* para resolução dos mais variados problemas [Oliveira 2007b]. Devido a isso, existe um grande número de funções que podem ser utilizadas como *wavelets*-mãe. No entanto, para desenvolvimento deste trabalho foi utilizado a família de *wavelets* de Haar.

O sistema de Haar é diversas vezes considerado como o mais natural e simples exemplo de um sistema ortonormal *wavelet* e que tem desempenhado um importante papel até os dias de hoje [Feichtinger 2006].

A função $\psi(t)$, denominada de função wavelet de Haar, é definida por:

$$\psi(t) = \begin{cases} 1, & 0 \leq t < 1/2, \\ -1, & 1/2 \leq t < 1, \\ 0, & \text{caso contrário} \end{cases} \quad (1)$$

Como as funções base de Haar são funções de passos descontínuos, sua utilização é adequada para as séries temporais que possuem variações bruscas de valores no decorrer do tempo. Ela também pode ser considerada uma “*wavelet* digital”, pois, sua forma de onda se parece com um pulso digital [Oliveira 2007b].

A escolha do tratamento dos dados foi feita empiricamente, onde as transformadas *wavelets* (famílias: dB, *coiflet*, *symlet*, e Haar) nível intermediário 3, foram aplicadas aos dados no tempo, e inseridas em uma rede Perceptron Multicamadas (PMC), com 2 camadas escondidas, 9 e 7 neurônios respectivamente, taxa de aprendizagem de 0,05, com 70 % dos dados para treinamento e 30 % para testes, sendo que ao fim a transformada *wavelet* Haar apresentou os melhores resultados.

2.3. Classificação de padrões sem otimização

Após a realização dos testes para encontrar a transformada *wavelet* (WT) com melhor resultado para redes neurais, novos testes foram realizados para busca da melhor topologia pelo método convencional (empírico).

Para garantir uma melhor classificação, separou-se as 9 subclasses em 3 grupos:

- (G1) estão as subclasses S1 - Orientação 1, nível de força fraco; S5 - Orientação 2, nível de força médio e S9 - Orientação 3, nível de força forte;
- (G2) estão as subclasses S2 - Orientação 1 nível de força médio; S6 - Orientação 2, nível de força forte e S7 - Orientação 3, nível de força fraco;
- (G3) estão as subclasses S3 - Orientação 1 nível de força forte; S4 - Orientação 2, nível de força fraco e S8 - Orientação 3, nível de força médio.

Foi aplicada uma sobreposição de informações, para aplicação da *wavelet* de 50%. Gerou-se assim, para cada uma das subclasses, 2928 dados. Para cada um dos grupos portanto, haviam 8784 amostras de dados. Nos testes foram utilizados 70% (6148) destes dados para treinamento e 30% (2636) para validação.

Para cada um dos grupos foi utilizado uma RNA, 12 testes foram realizados aplicando redes com diferentes topologias para cada um dos grupo de dados, alterando-se números de camadas, quantidade de neurônios por camada escondida, taxa de aprendizagem e confiabilidade. O algoritmo de treinamento empregado nestes experimentos foi o de Levenberg-Marquadt (LM). Todos os 12 testes são descritos na tabela 1.

Tabela 1. Testes - sem otimização de topologias

Nº do Teste	Nº Camadas escondidas	Topologia	Taxa de Aprendizado	Confiabilidade
1	2	[9 7]	0,1	0,9
2	2	[9 7]	0,05	0,9
3	2	[9 7]	0,1	0,98
4	2	[9 7]	0,05	0,98
5	3	[9 7 2]	0,1	0,9
6	3	[9 7 2]	0,05	0,9
7	3	[9 7 2]	0,1	0,98
8	3	[9 7 2]	0,05	0,98
9	4	[7 5 4 2]	0,1	0,9
10	4	[7 5 4 2]	0,05	0,9
11	4	[7 5 4 2]	0,1	0,98
12	4	[7 5 4 2]	0,05	0,98

2.4. Sistema de Otimização

Para realizar a implementação desta ferramenta de otimização, foi necessário primeiramente escolher os atributos a serem otimizados, bem como a forma com a qual os mesmos seriam representados. Cada topologia de RNA foi modelada como sendo um indivíduo da população do AG. Estes foram representados como sendo um vetor, onde os números representando a quantidade de neurônios nas camadas escondidas são inteiros, e aquele que representa a taxa de aprendizagem da rede como sendo de ponto flutuante, como se segue na figura 1. Vale destacar ainda que algumas restrições foram estabelecidas empiricamente para as topologias da RNA, como por exemplo: o número de neurônios nas camadas escondidas varia de 0 a 31, a taxa de aprendizado de 0.025 a 0.5, e o número de camadas escondidas de 1 a 3.

Durante a execução do AG, existe ainda outra representação dos indivíduos da população, que é usada especialmente nas fases de seleção, cruzamento (*crossover*) e

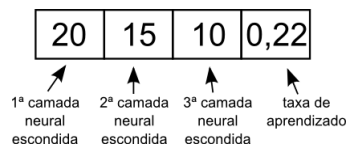


Figura 1. Representação de um indivíduo em decimal e float

mutação. Antes de passar por essas fases, os indivíduos são todos convertidos para a base binária. Esta forma de representação foi escolhida pois, por se tratar de uma ferramenta com propósitos educacionais para a utilização em conjunto com algoritmos elaborados em atividades práticas de RNAs, esta seria uma maneira didática de se modelar a população do AG, e portanto, mais compreensível a todos. A conversão ocorre da seguinte forma:

- Número de neurônios nas camadas escondidas: cada um destes números é convertido para a base binária de forma simples utilizando 5 bits, o que é suficiente para cobrir a faixa de 0 a 31.
- Taxa de aprendizado: Por ser um número de ponto flutuante, foi necessário elaborar um mapeamento mais avançado para realizar a sua conversão para a base binária. Para chegar na formulação matemática da mudança de base foi utilizada a teoria dos segmentos proporcionais. No caso deste trabalho, o ponto flutuante é mapeado para um número inteiro que pode ser representado com até 10 bits, e então esse número inteiro é convertido para binário, resultando na representação binária para aquele ponto flutuante.

Feita esta troca de base numérica, o resultado é uma cadeia de bits que representa um indivíduo da população (uma topologia de uma RNA), no formato apresentado na figura 2. Esta é a representação binária do indivíduo da figura 1.

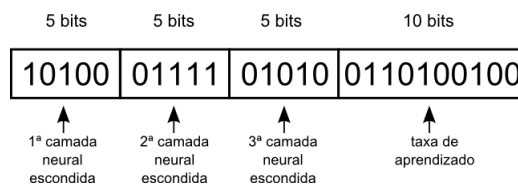


Figura 2. Representação de um indivíduo em binário

Após uma breve descrição das duas representações possíveis dos indivíduos utilizadas no AG, pode-se falar também sobre como o SIH foi desenvolvido, e qual é o fluxo de informações no mesmo. A figura 3 demonstra de forma simplificado o funcionamento do sistema.

De forma mais detalhada, as etapas do funcionamento do SIH desenvolvido são:

- Geração de topologias aleatórias: No início do algoritmo, é gerada apenas uma vez a chamada população inicial do AG, ou a primeira geração.
- Cálculo do *Fitness*: Nesta etapa, utilizando os dados de entrada, será feito o treinamento e validação da RNA, resultando numa porcentagem de acerto na classificação de padrões, que no caso deste trabalho é a função escolhida como *fitness*.

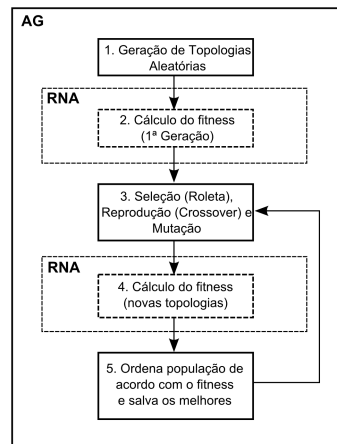


Figura 3. Diagrama do Sistema Inteligente Híbrido

- Seleção, Reprodução e Mutação: Utilização de um algoritmo de seleção (no caso deste trabalho, o algoritmo da roleta, que seleciona proporcionalmente à aptidão dos indivíduos) para selecionar aqueles indivíduos que se reproduzirão. Além disso, aplicação dos operadores de *crossover* e mutação nos indivíduos escolhidos na etapa de seleção. Durante o *crossover*, o ponto de cruzamento é escolhido aleatoriamente na cadeia de bits, que representa o indivíduo da população.
- Ordenação da população e preservação dos melhores: Nesta etapa, a população é ordenada de acordo com a aptidão e a complexidade do indivíduo.

Após detalhar as etapas de funcionamento do SIH em questão, é importante ainda frisar algumas estratégias empregadas na implementação deste sistema. A primeira delas é que, devido à grande variação nos resultados obtidos à cada vez que se realizava o treinamento da RNA com uma determinada topologia, optou-se por realizar-se pelo menos 3 e no máximo 5 treinamentos para cada topologia, e ao final utilizando-se apenas o melhor resultado obtido. Outra estratégia empregada foi de salvar a variável do tipo *network* que representa a melhor rede treinada para uma dada topologia e que obteve o melhor desempenho nos treinamentos. Isto foi feito pois a intenção do sistema é retornar de fato uma rede já treinada de forma otimizada. Por último, também foi usada uma tática visando priorizar as topologias que representariam uma complexidade computacional mais baixa na realização dos treinamentos das redes, ou seja, aquelas que possuem um número menor de camadas escondidas, ou ainda um número menor de neurônios nestas camadas.

Os testes realizados com o sistema otimizado são apresentados na tabela 2. No AG, foram utilizados os valores de 70 % para a taxa de *crossover*, 0,5% para a taxa de mutação e uma população de tamanho 20. Nestes experimentos ainda foi usada uma confiabilidade de 0,98 e divisão de 70% da base de dados para treinamento e 30% para validação da RNA, e o algoritmo de treinamento da rede empregado foi o *Scaled Conjugate Gradient backpropagation (SCGBP)*.

2.5. Equipamentos utilizados

Os testes do presente trabalho foram todos realizados em computadores com a mesma configuração, todos equipados com o processador Intel Core i5 3330 de 4 núcleos e 4 GB de memória RAM.

Tabela 2. Testes realizados no SIH

Testes do Sistema Otimizado		
TxCross = 0,7 - TxMut = 0,05		
TamPop = 20 - Confiabilidade = 0,98		
Nº Teste	Subclasses	Nº Gerações (AG)
13	G1	10
14	G1	25
15	G2	10
16	G2	25
17	G3	10
18	G3	25

3. Resultados

3.1. Experimentos realizados e resultados obtidos

Com os 12 testes realizados para cada um dos 3 grupos de subclasses, foi possível construir o gráfico apresentado na figura 4, onde, no eixo vertical, tem-se a taxa de acertos totais e, no eixo horizontal, o teste realizado.

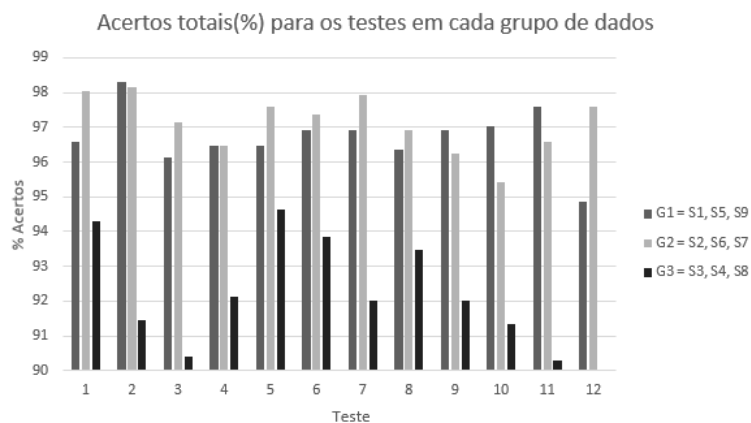


Figura 4. Gráfico com as taxas de acerto do sistema não otimizado

Com o gráfico, pode-se observar que o teste 2 possui os melhores resultados para os grupos 1 e 2. Porém, para o grupo 3, o melhor resultado é apresentado no teste 5. As matrizes de confusão mostradas na figura 5 apresentam o quanto cada uma das melhores redes encontradas classificam as subclasses.

Já os resultados obtidos nos testes com o sistema de otimização são apresentados nas matrizes de confusão da figura 6.

A tabela 3 demonstra o tempo de execução de cada um dos testes do sistema híbrido. A mesma também mostra a topologia da melhor rede obtida ao final da execução do AG.

Subclasses 1,5 e 9	Classificação (%)				
		1	5	9	Incerteza
Desejado	1	99,32	0,34	0	0
	5	0,68	98,29	0,34	0,34
	9	0,68	1,37	97,26	0,68
Total de acertos:		98,29			

Subclasses 2,6 e 7	Classificação (%)				
		2	6	7	Incerteza
Desejado	2	96,92	0,34	0,68	1,71
	6	0	98,97	0,34	0,68
	7	0,34	0,68	98,63	0,34
Total de acertos:		98,17			

Subclasses 3,4 e 8	Classificação (%)				
		3	4	8	Incerteza
Desejado	3	95,21	0	3,42	1,03
	4	0,34	95,21	3,08	1,03
	8	3,08	2,74	93,49	0,68
Total de acertos:		94,63			

Figura 5. Matrizes de confusão para os testes não otimizados

Tabela 3. Tempo de execução do SIH e Topologia das melhores redes para cada grupo

Sistema Inteligente Híbrido - Tempo e Melhores topologias				
Nº Teste	Subclasses	Nº Gerações (AG)	Topologia da melhor rede	Tempo Total (h)
13	G1	10	[25 31 4 0,45]	31:15:36
14	G1	25	[24 30 4 0,25]	75:48:00
15	G2	10	[30 17 5 0,07]	31:45:36
16	G2	25	[25 31 4 0,08]	80:25:48
17	G3	10	[30 22 5 0,33]	34:55:48
18	G3	25	[30 25 4 0,02]	81:28:48

3.2. Análise dos resultados

Utilizando-se da tabela 4, que demonstra um resumo dos resultados obtidos, pode-se comparar o resultado entre a classificação de padrões sem otimização e com otimização, assim como realizar a comparação entre quantidades de gerações.

Tabela 4. Comparação Otimizado vs Não-Otimizado

Subclasses	Sem Otimização (%)	Com Otimização (%)	Com Otimização (%)
		(10 Gerações)	(25 Gerações)
G1	98,29	99,96 (+1,67)	99,96 (+1,67)
G2	98,17	99,92 (+1,75)	99,96 (+1,79)
G3	94,63	99,35 (+4,72)	99,73 (+5,1)

Observando-se os valores obtidos nas classificações, nota-se que mesmo empregando um algoritmo de treinamento da rede com desempenho inferior (SCGBP) com relação ao LM, houve uma evolução na porcentagem de acertos da rede quando foi empregado o sistema híbrido (AG+RNA) em comparação ao classificador de padrões simples.

Subclasses 1,5 e 9	Classificação (%) - 10 Gerações				
		1	5	9	Incerteza
Desejado	1	100,00	0	0	0
	5	0	99,89	0	0,11
	9	0	0	100,00	0
Total de acertos:		99,96			

Subclasses 1,5 e 9	Classificação (%) - 25 Gerações				
		1	5	9	Incerteza
Desejado	1	100,00	0	0	0
	5	0	99,89	0	0,11
	9	0	0	100,00	0
Total de acertos:		99,96			

Subclasses 2,6 e 7	Classificação (%) - 10 Gerações				
		2	6	7	Incerteza
Desejado	2	99,89	0	0	0,11
	6	0	99,89	0	0,11
	7	0	0	100,00	0
Total de acertos:		99,92			

Subclasses 2,6 e 7	Classificação (%) - 25 Gerações				
		2	6	7	Incerteza
Desejado	2	100,00	0	0	0
	6	0	99,89	0	0,11
	7	0	0	100,00	0
Total de acertos:		99,96			

Subclasses 3,4 e 8	Classificação (%) - 10 Gerações				
		3	4	8	Incerteza
Desejado	3	99,89	0	0	0,11
	4	0	99,20	0	0,8
	8	0	0,12	98,96	0,92
Total de acertos:		99,35			

Subclasses 3,4 e 8	Classificação (%) - 25 Gerações				
		3	4	8	Incerteza
Desejado	3	99,89	0	0	0,11
	4	0	99,77	0	0,23
	8	0,11	0,11	99,55	0
Total de acertos:		99,73			

Figura 6. Matrizes de confusão para os testes otimizados

Já quando atenta-se ao comparativo entre diferentes números de gerações, percebe-se que no caso específico deste trabalho, com os dados de sinais eletromiográficos, não houve melhora significativa que justificasse o uso de mais gerações na otimização, principalmente quando avalia-se o tempo total de execução do SIH, apresentado na tabela 3. O SIH, com 25 gerações, gastou mais do que o dobro de tempo dos experimentos com 10 gerações, e melhorou de forma pouco significativa os resultados.

4. Considerações Finais

Fundamentando-se naquilo que foi observado na seção anterior, é possível concluir que a otimização de topologias de RNAs por meio do uso de AGs é uma abordagem que pode trazer melhorias para o desempenho da rede na classificação de padrões, e que portanto, a principal contribuição deste trabalho é uma ferramenta que poderá auxiliar tanto estudantes quanto professores na elaboração de algoritmos em atividades práticas de RNAs. Isso pode ser observado principalmente nos experimentos com as subclasses 3, 4 e 8, onde o aumento na porcentagem de acerto chegou a 4,72 % no AG com 10 gerações, e a 5,1 % no AG com 25 gerações.

Embora tenham sido identificadas melhorias no desempenho das RNAs quando comparados o sistema sem otimização com o SIH, a diferença entre as topologias ótimas encontradas com 10 e 25 gerações do AG não foi significativa o suficiente para justificar o uso do SIH com o banco de dados de sinais eletromiográficos com muitas gerações, já que o custo computacional aumenta muito com o número de gerações, algo que também pode ser constatado neste trabalho.

Conforme apresentado, a combinação de ferramentas inteligentes na forma de um SIH mostrou-se promissora na resolução do problema de classificação de movimentos, à medida que insere método na forma como os processos de se encontrar as melhores topologias de RNA tornam-se otimizados e não mais empíricos. A continuidade desta

pesquisa pode ser direcionada na utilização de técnicas de computação paralela, que reduzam o tempo computacional introduzido pela utilização do AG.

Referências

- Al-Timemy, A. H., Khushaba, R. N., Bugmann, G., and Escudero, J. (2015). Improving the performance against force variation of emg controlled multifunctional upper-limb prostheses for transradial amputees. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*.
- Belov, M. P. and Zolotov, O. I. (2015). Optimization of parameters of neural networks by genetic algorithm in the control systems of electromechanical objects. *2015 XVIII International Conference on Soft Computing and Measurements (SCM)*.
- Castilho, J., Domingues, M., Pagamisse, A., and Júnior, O. M. (2012). Introdução ao mundo das wavelets. *Notas em Matemática Aplicada*; v. 62.
- Çam, Z. G., Çimen, S., and Yıldırım, T. (2015). Learning parameter optimization of multi-layer perceptron using artificial bee colony, genetic algorithm and particle swarm optimization. *IEEE 13th International Symposium on Applied Machine Intelligence and Informatics*.
- Correa, A., González, A., and Ladino, C. (2011). Genetic algorithm optimization for selecting the best architecture of a multi-layer perceptron neural network: A credit scoring case. *SAS Global Forum 2011*.
- da Silva Severo, D. (2010). Otimização global em redes neurais. Trabalho de Conclusão de Curso.
- Feichtinger, H. (2006). Precursors in mathematics: Early wavelet bases. *Fundamental papers in wavelet theory*. Princeton: Princeton University Press, p. 149-154.
- Fermino, M. A., Pavoni, H., Spatti, D. H., and Dajer, M. E. (2016). Reconhecimento de Distúrbios Vocais Utilizando Redes Neurais Artificiais. *XXV Congresso Brasileiro de Engenharia Biomédica, 2016, Foz do Iguaçu. Anais do XXV CBEB - Edição 2016, 2016. p. 1825-1828*.
- Herle, S., Man, S., Lazea, G., Marcu, C., Raica, P., and Robotin, R. (2010). Hierarchical myoelectric control of a human upper limb prosthesis. *Robotics in Alpe-Adria-Danube Region (RAAD)*.
- Ivanov, O. and Gavrilau, M. (2014). Multilayer perceptron architecture optimization for peak load estimation. *2014 12th Symposium on Neural Network Applications in Electrical Engineering (NEUREL)*.
- Kamiński, M. (2016). Neural estimators of two-mass system optimized using the levenberg-marquardt training and genetic algorithm. *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*.
- Khushaba, R. N., Al-Timemy, A., Kodagoda, S., and Nazarpour, K. (2016). Combined influence of forearm orientation and muscular contraction on emg pattern recognition. *Expert Systems with Applications*.
- Oliveira, H. (2007a). *Análise de Fourier e Wavelets: Sinais Estacionários e não Estacionários*. EDUFPE.

- Oliveira, H. (2007b). *Análise de Sinais para Engenheiros: Uma Abordagem via Wavelets*. Brasport.
- Oskoei, M. and Hu, H. (2007). Myoelectric control systems - a survey. *Biomedical Signal Processing and Control*, vol. 2, no. 4.
- Ramchoun, H., Amine, M., Idrissi, J., Ghanou, Y., and Ettaouil, M. (2016). Multilayer perceptron: Architecture optimization and training. *International Journal of Interactive Multimedia and Artificial Intelligence*.

Criptografia de Fluxo por Autômatos Celulares: uma abordagem baseada na combinação de Busca Tabu com *Simulated Annealing*

André de Souza Brito, Stênio Sã Rosário Furtado Soares, Saulo Moraes Villela

¹Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora
36036-900 – Juiz de Fora, MG – Brasil

andre.brito@ice.ufjf.br, {stenio.soares, saulo.moraes}@ufjf.edu.br

Abstract. *This paper describes the development of an algorithm to generate stream cypher keys based on one-dimensional chaotic cellular automata combined with the metaheuristics Simulated Annealing and Tabu Search. We introduce the main concepts used in the algorithm construction and then the results obtained. Finally, we discuss the application of this approach on real problems.*

Resumo. *Esse trabalho descreve o desenvolvimento de um algoritmo para geração de chaves criptográficas de fluxo baseadas em autômatos celulares caóticos unidimensionais combinados às meta-heurísticas Simulated Annealing e Busca Tabu. Inicialmente são apresentados os principais conceitos utilizados na construção do algoritmo e depois os resultados obtidos. Por fim, é discutida a eventual aplicação deste tipo de abordagem em problemas de natureza real.*

1. Introdução

É consenso no meio científico que, no contexto atual em que os sistemas estão cada vez mais conectados via rede, a segurança de informação é requisito essencial destes sistemas. Grande parte dos equipamentos e sistemas possuem mecanismos que garantem, por exemplo, que dados estejam seguros durante uma transação bancária ou em serviços de telefonia celular. Neste cenário, vale destacar um conjunto de técnicas que seguem um mesmo princípio de funcionamento conhecido como criptografia. A criptografia garante a segurança das informações ao escondê-las, tornando-as legíveis apenas ao destinatário.

A criptografia pressupõe a aplicação de uma chave sobre a mensagem, escondendo-a no envio ou revelando-a ao chegar no destinatário. O sucesso de um algoritmo desse tipo depende diretamente da natureza da chave usada pelo mesmo. Muito do esforço usado na criação desses sistemas é direcionado para a tarefa de garantir que a chave usada assegure a segurança dos dados que foram criptografados.

Em algumas situações não é possível ter de antemão acesso a todo o conteúdo a ser criptografado. Os algoritmos que lidam com esse tipo de situação são chamados algoritmos criptográficos de fluxo e usam algum artifício para contornar essa restrição. Uma forma simples de contornar o problema é criptografar o dado que estiver disponível no momento. Para isso, algumas abordagens usam um elemento chamado gerador, capaz de manter a geração de uma chave segura, enquanto houver dados sendo transmitidos. Criar um gerador eficiente é uma tarefa árdua e uma grande variedade de métodos já foi estudada com esse propósito [Zeng et al. 1991]. Um desses métodos consiste na utilização de

estruturas discretas com comportamento evolutivo conhecidas como Autômatos Celulares (ACs), que são aplicáveis a esse domínio por incorporar complexidade ao processo e serem de fácil implementação.

Wolfram abriu um novo horizonte no campo criptográfico, ao definir as diretrizes iniciais do uso de autômatos celulares como mecanismos fundamentais de dispositivos criptográficos [Wolfram 1985]. O autor apresenta um estudo sobre a segurança de sistemas fundamentados na utilização de ACs como mecanismos geradores. A segurança destes sistemas está principalmente na dificuldade de, em um dado instante de tempo, descobrir a configuração atual do AC, dada a complexidade das interações entre os elementos no decorrer das gerações das chaves.

Tomassini e Perrenou, por exemplo, propuseram uma abordagem evolucionista que combina Algoritmos Genéticos e ACs [Tomassini and Perrenou 2001]. Os autores defendem que a simples combinação de determinados parâmetros de um AC já seria capaz de assegurar a segurança dos dados criptografados. Villela e Carvalho analisaram extensivamente os problemas em se utilizar a solução sugerida por Tomassini e Perrenou e propuseram a utilização da técnica de *Simulated Annealing* – SA, com foco na geração de chaves de fluxo baseadas em ACs com características especiais, conhecidos como ACs caóticos [Villela and Carvalho 2007]. Embora ACs caóticos sejam altamente aplicáveis a sistemas criptográficos, os ACs usados apresentavam limitações severas em relação ao tamanho da chave gerada, tornando-os ineficazes para aplicações reais.

Esse trabalho apresenta uma abordagem que combina as metaheurísticas *Simulated Annealing* e Busca Tabu para a geração de ACs caóticos aplicáveis a algoritmos criptográficos de fluxo. Nas Seções 2 e 3 são apresentados os aspectos teóricos do problema e de ACs. Nas Seções 4, 5 e 6 são descritas as metaheurísticas desenvolvidas e a combinação das mesmas. Na Seção 7 são apresentados os resultados computacionais e a análise da eficácia das abordagens propostas através de testes consolidados na literatura e, por fim, é apresentada uma análise do efeito das alterações propostas no decorrer do processo de desenvolvimento dos algoritmos.

2. Criptografia de fluxo

Algoritmos de fluxo são algoritmos criptográficos ditos algoritmos de chave simétrica. Esses algoritmos usam chaves criptográficas fortemente relacionadas para os processos de encriptação e desencriptação. Na prática, uma única chave pode ser usada para os dois processos e, na maioria dos casos reais, é isso que ocorre.

Em criptografia de fluxo, todas as operações são realizadas em bits individuais da mensagem. O processo *one-time pad* [Miller 1882] consiste na combinação de cada bit da mensagem com um bit de uma sequência aleatória, isto por meio de uma operação matemática L . A operação de desencriptação pode ser realizada basicamente aplicando a operação inversa L_{inv} aos bits da mensagem cifrada e aos da sequência aleatória.

A sequência aleatória usada em todo o processo é denominada chave de fluxo. Esta chave deve ter o mesmo número de bits da mensagem original e ser formada por bits que variam no tempo. Assim, cada bit da mensagem é criptografado por um bit diferente da chave. Um problema operacional é gerado ao utilizar-se de uma chave de fluxo totalmente aleatória. Como a chave usada é a mesma para o processo de encriptação e desencriptação (Figura 1), ela deve ser armazenada e repassada ao receptor, o que pode comprometer a

segurança do algoritmo. Para contornar este problema, utiliza-se uma sequência de bits pseudoaleatória, que pode ser reconstruída caso necessário.

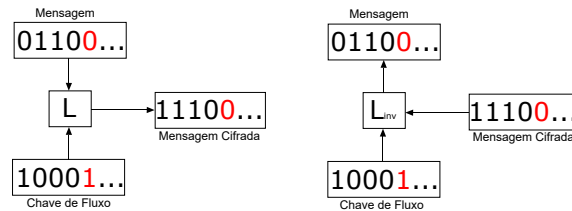


Figura 1. Processos de encriptação (esquerda) e descriptação (direita).

Alguma máquina, ou algoritmo determinístico, deve ser implementado em conjunto com o algoritmo de fluxo para gerar a chave pseudoaleatória. Este elemento gerador deve ser capaz de reconstruir a chave quando necessário e garantir que ela tenha características aleatórias o suficiente para ser considerada segura. Esses algoritmos, em geral, têm um elemento disparador conhecido como chave semente, que é responsável por calibrar os parâmetros do gerador e ativar uma chave de fluxo específica dentro do mesmo. O emissor e o receptor, portanto, só precisam compartilhar a chave semente. Uma representação conceitual de como se dá o processo é mostrada na Figura 2.

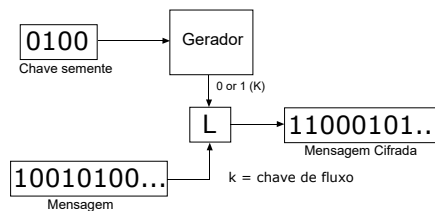


Figura 2. Processo de encriptação usando uma operação L e um gerador.

3. Autômatos celulares

Um autômato celular (AC), em sua essência, é um modelo matemático capaz de simular os mais diversos fenômenos do espaço partindo de um conjunto de regras locais simples [Wolfram 1982]. Um AC é composto por um conjunto de elementos menores (células) distribuídos em uma rede uniforme n -dimensional. Cada célula possui um valor atribuído que representa um dentre os vários estados possíveis na formulação do problema.

Um estado é definido pela aplicação de uma regra estática, que geralmente depende do seu estado anterior e do estado anterior de suas células adjacentes (vizinhança). Essa dependência causa uma propagação de efeitos que, junto das regras de transição, dá o caráter evolutivo dos ACs. A cada instante de tempo, uma nova geração de células é concebida pela aplicação da regra de transição correspondente a cada célula de forma independente. Com isso, pode-se definir um AC por seus quatro componentes principais:

- Um conjunto de valores possíveis aos estados das células do autômato;
- Um estado inicial para as células do autômato;
- Uma vizinhança, que pode variar com a dimensão ou forma de modelar o problema. Em um AC unidimensional, por exemplo, uma célula é ligada localmente a $2r + 1$ vizinhos, onde r (raio) é o número de células vizinhas (Figura 3);

- Um conjunto de regras de transição que permite calcular o estado das células com base no seu estado anterior. Uma regra é, basicamente, uma máquina de estados finitos, que mapeia uma transição para cada configuração possível de vizinhança (Figura 4). O conjunto de regras associado a um AC pode conter somente uma regra ou uma regra associada a cada célula. O autômato correspondente a cada caso é chamado, respectivamente, de homogêneo ou heterogêneo.

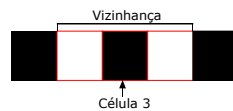


Figura 3. Vizinhança de raio 1 da célula 3 de um AC unidimensional.

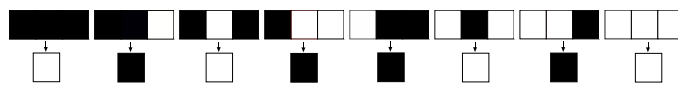


Figura 4. Exemplo de regra de transição (regra 90).

Segundo Wolfram, apesar das diferentes regras e configurações iniciais, os autômatos têm comportamentos que permitem catalogá-los em 4 classes [Wolfram 2002], como mostra a Figura 5: (i) aqueles que, a partir de um estado inicial qualquer, todas as células convergem para um mesmo estado; (ii) aqueles que geram estruturas estáveis ou periódicas muito simples; (iii) os que geram estruturas com comportamento caótico por um número suficiente de gerações; (iv) aqueles que possuem um comportamento complexo, gerando estruturas espaço-temporais que em raros casos podem ser persistentes.

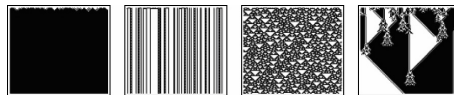


Figura 5. Classes observadas por Wolfram: (i)-(iv) – da esquerda para direita.

4. Busca Tabu

Busca Tabu é um procedimento adaptativo que guia o algoritmo de busca local na exploração do espaço de busca ao mesmo tempo que evita retornar a ótimos locais já visitados [Glover 1986]. O algoritmo de Busca Tabu explora, a cada iteração, a vizinhança da solução corrente, movendo-se na direção do melhor vizinho ao tempo que evita soluções já visitadas (Tabus) a partir do registro destas em uma lista restritiva. Cada nova solução fica armazenada na lista restritiva (lista Tabu), que contém as últimas soluções visitadas pelo algoritmo. Isso evita ciclos e permite uma melhor exploração do espaço de soluções. O algoritmo é executado até que soluções melhores parem de ser geradas ou até que a melhor solução chegue a um limite definido.

A lista Tabu funciona como uma memória para o algoritmo. A Busca Tabu tem dois tipos de memórias diferentes. A memória de curta duração armazena as melhores soluções por um curto período de tempo, fazendo com que, eventualmente, uma solução tenha que ser retirada para dar lugar a outra. A solução retirada geralmente é a mais antiga. Já a memória de longa duração, por poder armazenar um número maior de soluções,

armazena essas mesmas soluções por um período maior de tempo. Na maior parte dos casos, a tarefa de manipular soluções dentro da lista Tabu é inviável ou bastante complexa. Usa-se, então, o artifício de armazenar somente características das soluções na lista (movimentos). Esses movimentos vão de simples alterações de bits em uma configuração binária até atribuições mais complexas, como permutações em vértices de um grafo.

O problema desta abordagem é que soluções já vistas, ou eventuais soluções geradas a partir da mesma, não serão analisadas. Porém, esta limitação pode ser contornada, já que o caráter Tabu de alguns movimentos pode ser simplesmente ignorado. Assim, uma função especial (função de aspiração) é utilizada no sentido de permitir que um movimento Tabu seja realizado, por exemplo, se houver uma melhora no valor da função objetivo ou se todos os movimentos disponíveis forem Tabus.

5. Simulated Annealing

A técnica de *Simulated Annealing* (SA) é uma metaheurística que usa o conceito de energia associado à qualidade de cada solução obtida. Alterações em uma solução podem levar a um aumento ou diminuição dessa energia. O objetivo é achar a melhor solução através da maximização de uma função associada a essa energia [Kirkpatrick et al. 1983].

O algoritmo inicia com uma solução s qualquer e, a cada iteração, s é substituída por uma solução obtida na sua vizinhança, o que pode provocar uma variação na energia do sistema. Alterações que levem a um aumento na energia do sistema são sempre aceitas, enquanto as que geram redução são aceitas com uma probabilidade p , dada por $p = e^{-\frac{d_z}{KT}}$, onde d_z é a diferença de energia associada e T é a temperatura atual. O valor K é um valor conhecido como constante de Boltzmann. Inicialmente, a temperatura T assume um valor alto (T_0) e decai ao longo da execução. Para cada temperatura são geradas soluções por um número definido de iterações. O algoritmo deve convergir para o equilíbrio térmico, sendo que a temperatura é diminuída até chegar a um estado de alta energia por meio da equação: $T_n = \alpha T_{n-1}$, onde T_n é a temperatura em um estágio de resfriamento n e α é um fator associado à estratégia de resfriamento.

O procedimento é finalizado quando a temperatura chega a um valor próximo de zero, que indica um estado de “cristalização” do sistema. Neste estado, qualquer alteração realizada não provocará um aumento na função objetivo e nenhuma solução com o valor pior que a melhor solução pode ser aceita. Espera-se que neste estado a melhor solução obtida esteja próxima a um ótimo global.

5.1. Entropia

Na termodinâmica, a entropia é uma grandeza associada à desordem de um sistema físico. Uma variação de entropia no sistema ocorre em situações onde há transferência de energia na forma de calor. Isto acontece porque a energia transferida nunca é totalmente transformada em trabalho. Assim, uma parte dessa energia dissipada é transformada em entropia, aumentando a desordem do sistema. Na teoria da informação, a entropia está relacionada à quantidade de informação necessária para compreender um dado fenômeno [Shannon 1948]. Quando não há como prever o comportamento do fenômeno, a análise é feita de forma probabilística, considerando a probabilidade p_i à ocorrência de cada evento E_i . A entropia de informação é a quantidade média de informação por evento:

$$H = - \sum_{i=1}^n p_i \log_2 p_i. \quad (1)$$

O máximo da equação (1) representa um fenômeno com um grande número de variáveis (eventos) de difícil previsão. Portanto, pode-se afirmar que a entropia de informação de um determinado fenômeno está diretamente relacionada à quantidade de eventos e à dificuldade de prevê-los. De forma sucinta, a entropia da informação pode ser descrita como a falta de informação sobre um determinado assunto, o que causa a incerteza e a imprevisibilidade. Estas características fazem dessa grandeza um bom parâmetro para medir a qualidade de chaves geradas por sistemas criptográficos. Uma chave gerada nesses sistemas deve incorporar essa incerteza e essa imprevisibilidade.

6. Abordagens utilizadas

Nesse trabalho foram desenvolvidas abordagens baseadas na combinação das metaheurísticas *Simulated Annealing* e Busca Tabu para otimizar a escolha de autômatos celulares, para que sejam utilizados como mecanismos geradores de chave de fluxo.

Foram usados autômatos celulares unidimensionais binários heterogêneos, para adicionar complexidade ao processo, de 16 bits, que se aproximam mais de aplicações reais, e foram gerados a partir de regras aplicadas a uma vizinhança de tamanho 3 e de forma circular. Três versões do algoritmo principal foram desenvolvidas. A primeira usa o SA clássico puro; a segunda versão combina o SA com um mecanismo probabilístico para melhorar a escolha das regras durante as iterações; por fim, uma versão que combina o SA com o mecanismo probabilístico e a Busca Tabu.

6.1. SA Clássico

Duas versões do SA Clássico foram utilizadas. A diferença entre as mesmas está no critério adotado para verificação do equilíbrio térmico em uma dada temperatura. Na primeira versão, denotada “SA1”, o equilíbrio térmico ocorre quando a média aproximada da função objetivo chega a um estado onde as variações geradas são bem pequenas e próximas ao seu valor máximo e exato. Na segunda versão (“SA2”), o equilíbrio térmico é dado por um número de iterações do algoritmo sem melhora no valor da função objetivo.

A função objetivo utilizada foi a entropia, definida na equação (1), onde cada evento E_i representa uma subsequência diferente que pode ser gerada pelo autômato celular, e o valor n é o número máximo de subsequências diferentes que podem ser geradas a partir desse mesmo autômato. Este valor é calculado com base no tamanho $\|\tau\|$ e no número de estados que o automato usado pode assumir Q , num total de $Q^{\|\tau\|}$ variações possíveis. Para determinar quando o sistema “cristaliza”, é criado um ponto de temperatura mínima do sistema. O algoritmo é detalhado como segue:

1. É gerada uma solução inicial s_0 formada por um autômato celular e seu conjunto de regras correspondente, ambos gerados aleatoriamente.
2. Uma temperatura inicial elevada T_0 é associada a s_0 . É definido o critério utilizado para verificar o equilíbrio térmico em uma dada temperatura.
 - (a) Se o critério usado for o número de iterações sem melhora na função objetivo (SA2), é criado um contador $iterT$ para guardar esse número e outro valor $iterMax$ é definido como limite para $iterT$.
 - (b) Se o critério usado for a estabilização da média da função objetivo (SA1), é definido um número mínimo de iterações por temperatura.

3. O autômato é evoluído por x passos a partir da aplicação do conjunto de regras de transição. Este valor corresponde ao número máximo de subsequências que podem ser geradas sem repetição por um autômato do tamanho utilizado. Por fim, calcula-se a entropia a partir desse número de passos.
4. Uma das regras do autômato é sorteada aleatoriamente e trocada por outra (entre 0 e 255), gerando uma solução corrente s . O autômato é novamente evoluído, só que com essa nova regra no conjunto e a entropia é recalculada.
5. É calculada a variação Δ entre o valor da entropia calculada antes da regra ser alterada e o valor da entropia calculada depois que a regra foi alterada.
 - (a) Se $\Delta \geq 0$, s é aceita como nova solução a partir desse instante. O conjunto de regras do autômato passa a conter a regra alterada. Se o critério usado para equilíbrio térmico for o número de iterações sem melhora na função objetivo, o $iterT$ é zerado se Δ for diferente de 0 e incrementado se Δ for igual a 0.
 - (b) Se Δ for negativo, s pode ser ou não ser aceita como nova solução. Para determinar isso, é calculada a probabilidade $p = e^{-\frac{\Delta}{kT}}$. Um número aleatório (entre 0 e 1) é sorteado. Se o número cair dentro do intervalo entre 0 e p , a regra alterada permanece no conjunto de regras do autômato e s é aceita como nova solução. Caso contrário, o valor original da regra alterada é restabelecido e a solução s é descartada. O valor $iterT$ é incrementado se o critério usado para equilíbrio térmico for o número de iterações sem melhora na função objetivo.
6. Verifica-se se o sistema atingiu o equilíbrio térmico.
 - (a) No primeiro caso (SA1) é verificado se o valor de $iterT$ é igual ao valor $iterMax$.
 - i. Se o valor for igual, a temperatura atual tem um leve decréscimo dado por α e o algoritmo retorna ao passo 4.
 - ii. Se o valor for diferente, retorna-se ao passo 4 sem mudar a temperatura.
 - (b) No segundo caso (SA2) é necessário calcular a diferença das médias da entropia atual e da entropia há x (número mínimo de iterações por temperatura) atrás.
 - i. Se a diferença for menor que o valor ϵ , a temperatura sofre um pequeno decréscimo dado por α e o algoritmo retorna ao passo 4.
 - ii. Do contrário, retorna-se ao passo 4 sem mudar a temperatura.
7. O fim do processo ocorre quando a temperatura chega a um valor muito próximo de 0 (temperatura mínima), que indica que o sistema está cristalizado.

Os parâmetros usados descritos a seguir foram obtidos a partir de exaustivos testes: Temperatura inicial = 0,1; Decréscimo da temperatura (α) = 0,98; Temperatura mínima = 0,001; Mínimo de iterações por temperatura = 200; Diferença das médias (ϵ) = 0,01.

O número máximo de iterações ($iterMax$) é um parâmetro variável durante o algoritmo e tem seu valor inicial obtido a partir da equação $T_n = \alpha T_{n-1}$. Chegando a um valor $m = \log_{\alpha} \frac{T_{min}}{T_0}$, onde α , T_{min} e T_0 são, respectivamente, o decréscimo da temperatura, a temperatura inicial e a temperatura mínima usadas como parâmetro no SA. O valor inicial desse parâmetro foi definido como 300 iterações e ele é aumentado gradativamente até chegar a 600 iterações, quando tem-se temperaturas mais baixas e o algoritmo aplica busca local nas soluções.

6.2. SA Clássico e mecanismo probabilístico (Roleta)

Na abordagem clássica do SA, todas as regras têm uma mesma probabilidade na escolha. Porém, como já foi observado em vários trabalhos da literatura, e depois de um número suficiente de testes, verificou-se que só um pequeno número de regras leva a soluções

boas. Assim, um mecanismo probabilístico é usado para dar maior probabilidade de escolha às regras que levaram a resultados melhores em iterações ou execuções anteriores.

Para tanto, uma matriz bidimensional $i \times j$ foi criada. As linhas da matriz (i) são as regras presente no conjunto de regras e as colunas (j) o número de bits do autômato utilizado. Com isso, uma regra terá uma probabilidade diferente em cada célula do autômato. Visto que esse é um problema onde não só a regra é importante, mas também a posição onde essa regra aparece. A matriz armazena o número de vezes em que uma regra apareceu em uma determinada posição para um alvo definido. Este alvo tem um valor um pouco inferior ao valor máximo da função objetivo obtida até então na execução atual ou de uma execução anterior, quando a matriz é importada de outra execução. Todas as regras das soluções que estiverem no intervalo entre o valor alvo e o valor máximo da função objetivo são armazenadas na matriz na posição em que aparecem na solução. No cálculo da regra que será trocada em uma dada posição, a matriz é convertida em uma matriz de probabilidade e a função aleatória passa a dar mais peso à regra com maior probabilidade na posição informada, mecanismo denotado neste trabalho como Roleta.

A roleta permite que o algoritmo seja direcionado às melhores regiões do espaço de busca. A matriz é zerada toda vez que o valor da função objetivo é melhorado e o valor do alvo é menor que o valor referente a nova solução. Um novo alvo é definido por: $alvo = \frac{melhorEntropia}{maxEntropia} * \sigma$, onde $melhorEntropia$ é o valor máximo de entropia obtido até então, $maxEntropia$ é o máximo global da função de entropia e σ é o fator de proporcionalidade, fixado em 0,95.

Testes preliminares mostraram que esta abordagem melhora o tempo de execução e a convergência do algoritmo. Porém, em certos casos, principalmente em regiões com um número grande de soluções homogêneas ou próximo de muitos ótimos locais, a roleta vicia e o algoritmo é levado a convergir precocemente à valores distantes do ótimo.

6.3. SA Clássico e Roleta combinados com a Busca Tabu

Para minimizar o impacto da convergência precoce do algoritmo com o uso da Roleta, optou-se por inserir um módulo de Busca Tabu na abordagem. Com sua política de exploração extensiva da vizinhança de uma solução, objetivou-se sanar o problema. A ideia central é que a Busca Tabu atue periodicamente sobre soluções geradas pelo SA, como forma de evitar um aumento considerável na complexidade do algoritmo.

Na abordagem proposta, foram empregadas as ideias de memória adaptativa (lista Tabu) e estratégia de busca baseada em memória do algoritmo padrão da Busca Tabu. A lista Tabu usada suporta 100 movimentos. O tamanho da lista foi definido depois de diversos testes e os movimentos Tabu são aqui representados por trocas de regras em determinadas posições. Toda vez que a Busca Tabu é chamada, ao invés de sortear uma regra e uma posição baseada na Roleta, somente a regra é sorteada.

A partir de uma solução obtida, a vizinhança desta nova solução é analisada por meio da aplicação da regra sorteada em cada célula do autômato. Em cada troca, a entropia é calculada e armazenada em um vetor e a regra anterior é restabelecida à sua posição. Depois de todas as trocas, o vetor é ordenado de forma decrescente e a primeira posição que terá ao movimento que levou ao maior valor de entropia, dentre os calculados, é escolhida como solução corrente. O movimento para chegar a essa solução é armazenado na lista Tabu e se torna um movimento proibido.

O critério Tabu dos movimentos deve ser respeitado. Por isso, se o movimento escolhido como o melhor for um movimento proibido (Tabu), ele poderá ser descartado como solução e o próximo movimento, que representa o segundo maior valor obtido, será escolhido, e assim sucessivamente. O movimento Tabu não é sumariamente descartado, pois, se ele melhorar a função objetivo ou se todos os movimentos disponíveis forem Tabu, ele é escolhido mesmo assim (função de aspiração). O processo pode ser entendido com mais clareza a partir do seguinte algoritmo:

1. Uma regra é sorteada aleatoriamente (entre 0 e 255).
2. A regra sorteada é associada a uma célula do autômato.
3. A entropia é calculada usando o novo conjunto de regras e armazenada em um vetor.
4. A regra que foi trocada pela regra sorteada é devolvida à célula correspondente.
5. É feita a verificação se toda vizinhança da solução foi analisada.
 - (a) Se a vizinhança não foi toda analisada, o passo 2 do algoritmo é repetido para uma nova célula do autômato. Isso é feito porque há trocas a serem realizadas.
 - (b) Se toda a vizinhança foi analisada, o próximo passo pode ser realizado.
6. O vetor com as entropias é ordenado de forma decrescente. A solução correspondente ao movimento encontrado na primeira posição pode ser escolhida como solução corrente.
 - (a) O movimento é escolhido como solução corrente se não for um movimento Tabu, ou se os critérios da função de aspiração não forem atendidos (melhora na função objetivo ou se todos os movimentos forem Tabu).
 - (b) Se nenhuma dessas situações ocorrer, o passo 6 é repetido só que com a possível solução sendo a posição seguinte do vetor.
7. O movimento escolhido é adicionado à lista Tabu. Se a lista estiver cheia, o movimento mais antigo é retirado.
8. Por fim, a solução é passada ao SA.

7. Resultados Computacionais

Na comparação dos algoritmos, utilizou-se análise probabilística, que relaciona o tempo de processamento à probabilidade acumulada de convergência dos algoritmos. Assim, pode-se observar, em um determinado tempo de processamento, a probabilidade da convergência do algoritmo para um valor alvo preestabelecido para função objetivo.

No primeiro teste, foram comparadas as duas abordagens clássicas e as duas abordagens combinadas à Roleta. Os testes foram definidos sobre três alvos, que correspondem a 50%, 75% e 90% do valor máximo da entropia. Como já foi dito, a simples combinação entre Roleta e os SAs apresenta problemas em situações próximas a muitos ótimos locais e à regiões com muitas soluções homogêneas próximas a melhor solução.

Uma solução inicial foi gerada a partir de um autômato com configuração inicial $\{1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0\}$ e um conjunto de regras $\{91, 208, 189, 242, 223, 168, 200, 68, 162, 34, 115, 229, 155, 222, 114, 183\}$ cujo valor de entropia foi de 1,00125. Partindo desta solução, o algoritmo foi executado 50 vezes para os alvos de 50% e 75% e 20 vezes para o alvo de 90%. O algoritmo é parado quando o alvo é atingido e o tempo necessário é contabilizado.

A Figura 6-(a) mostra que, para obter soluções com entropia a 50% do ótimo, as duas abordagens que usam Roleta sempre têm chance de sucesso antes das abordagens puras. Ao tomar um alvo mais difícil, como o apresentado na Figura 6-(b), o comportamento se mantém, mas observa-se uma maior eficácia das abordagens com Roleta, que têm uma probabilidade maior de alcançar o alvo definido num intervalo de tempo menor.

Para a entropia em 90% como alvo, foi feita uma alteração no algoritmo, de forma que a Roleta só seja utilizada até que o algoritmo obtenha uma solução com entropia de 60% do valor máximo. Este valor foi definido observando testes empíricos e o gráfico da Figura 6-(b), onde é visto que as abordagens que combinam Roletas já começam a apresentar um comportamento deficiente a partir de um certo tempo, quando as curvas referentes às abordagens com Roleta ultrapassam as curvas referentes às abordagens puras, indicando que estas apresentam maior probabilidade de alcançar o alvo.

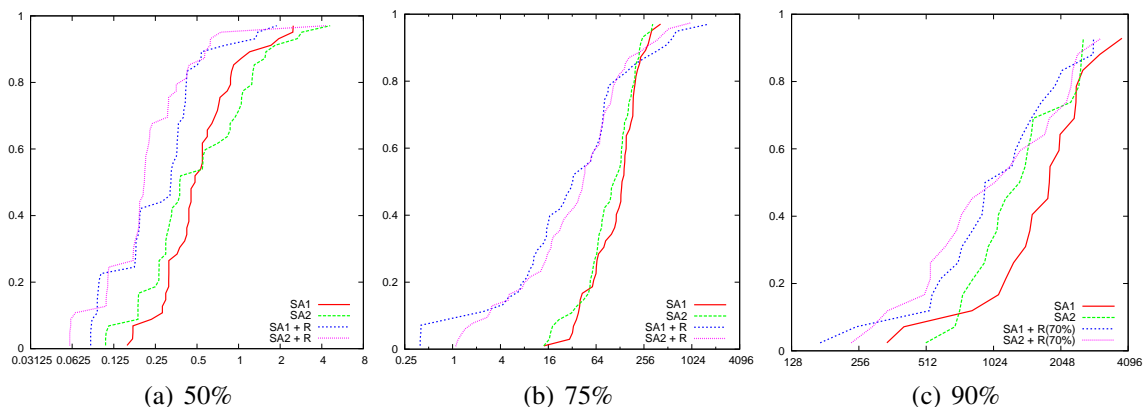


Figura 6. Gráficos probabilísticos (Convergência x Tempo)

Os gráficos das Figuras 6-(a), 6-(b) e 6-(c) evidenciam o que estava sendo defendido: a Roleta consiste em uma abordagem eficaz e pode melhorar o tempo do algoritmo. Para validar estas observações, foi feito um experimento fundamentado em 10 execuções independentes dos algoritmos. Os resultados obtidos são mostrados na Tabela 1.

Tabela 1. Valores de entropia das diferentes abordagens

Abordagem	Valor máximo	Valor médio
SA1	15,7533	14,2165 ± 1,0539
SA2	15,7599	14,1861 ± 0,7549
SA1 + R	15,2490	14,2031 ± 0,6126
SA2 + R	15,5099	14,0692 ± 1,0851

A análise da tabela serviu como base para definir qual versão desenvolvida seria mais adequada a ser combinada com a Busca Tabu. Entre as duas versões do SA, escolheu-se a primeira devido ao valor médio das soluções. Decidiu-se, também, utilizar a Roleta (até 60% do valor máximo), uma vez que os valores médios são bastante próximos e os tempos foram melhores, evidenciando a “partida inteligente”. A Busca Tabu foi anexada a essa versão (SA1 + R) como uma subrotina que é executada sempre que o algoritmo fica por um dado número de iterações sem conseguir uma melhora na solução. Esperava-se que essa nova combinação apresentasse resultados mais satisfatórios que das demais versões e isso foi o observado durante os testes realizados. Essa versão obteve como melhor solução um valor 15,9999 de entropia e um valor médio de entropia de $15,5483 \pm 0,2790$, resultado superior ao das abordagens descritas anteriormente.

A solução com valor de entropia máximo 15,9999, formada pelas regras {89, 101, 170, 149, 154, 149, 149, 86, 149, 89, 86, 86, 170, 170, 170, 166} e de configuração

inicial {0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0}, foi submetida à bateria de testes do NIST (*National Institute of Standards and Technology*) com o objetivo de avaliar o potencial criptográfico da chave gerada a partir da mesma, testando sua aleatoriedade. A versão utilizada possui 15 testes responsáveis por verificar a presença dos tipos de não-aleatoriedade mais comuns em uma sequência. Uma descrição completa dos testes pode ser encontrada em [Government 2011].

Foram executados 13 dos 15 testes. Os testes de Posto de Matriz Binária e Transformada Discreta de Fourier não foram executados, pois o número de bits da sequência não é adequado para esses testes. Para o primeiro, por exemplo, recomenda-se que o número de bits na entrada não ultrapasse 1.000.000 bits. Nos testes o valor obtido como retorno (p -valor) varia entre 0 e 1 e deve ser maior que 0,01 para que a sequência seja considerada aleatória. O tamanho da sequência gerada é de 1.048.256 bits. Os resultados dos testes são apresentados na Tabela 2 e mostram a robustez da proposta.

Tabela 2. Resultados dos testes do NIST

Teste	p -valor	Resultado
Entropia Aproximada	1,000000	Passou
Frequência de Blocos	0,474492	Passou
Somas Cumulativas	0,487832	Passou
Frequência	1,000000	Passou
Complexidade Linear	0,999967	Passou
Corrida Mais Longa	0,771896	Passou
Não Sobreposição de Padrão	0,010299 a 0,999866	Passou
Sobreposição de Padrão	0,701947	Passou
Excursões Aleatórias	0,235225 a 0,966457	Passou
Variante de Excursões Aleatórias	0,227711 a 0,977680	Passou
Corridas	0,781515	Passou
Serial	1,000000	Passou
Universal	0,691993	Passou

8. Considerações finais

A partir da análise dos resultados, ficou evidente a melhora proporcionada pela adição do método da Roleta ao SA clássico. Entretanto, observou-se a tendência deste método de aleatorização viciar, prejudicando consideravelmente a convergência do algoritmo quando submetido a alvos maiores que 75% do valor ótimo. Estes resultados motivaram a utilização de uma abordagem em que a Roleta é usada até uma certa distância percentual da solução ótima e depois é descartada, dando lugar a abordagem simples, funcionando como uma “partida inteligente” para o algoritmo.

Esta escolha permitiu ao algoritmo atingir alvos maiores, convergindo a resultados próximos ao ótimo com uma taxa maior que das abordagens clássicas. Para dar um incremento ainda maior nessa taxa de convergência foi incluído um módulo de Busca Tabu. Sendo esta variação a versão que obteve a melhor taxa de convergência e foi responsável pelas melhores soluções até então, alcançando soluções que variam de 98% a 99,5% do valor máximo da função de entropia correspondente a subsequências de 16 bits.

O problema da Busca Tabu está no tempo usado para chegar a essas soluções,

fazendo com que, em situações onde o tempo é importante, como em aplicações que o fluxo de informações é contínuo, esta abordagem não seja a mais adequada.

Como proposta de trabalhos futuros, sugere-se a utilização de outras metaheurísticas para o problema, como, por exemplo, o ILS (*Iterated Local Search*) e o VNS (*Variable Neighborhood Search*), que podem ser combinados às metaheurísticas já desenvolvidas ou usadas como novas abordagens.

Uma outra vertente a ser explorada é a possibilidade de calibrar melhor os parâmetros a serem usados nas versões desenvolvidas do algoritmo. Esta calibração, que atualmente é feita por meios de testes empíricos, pode ser realizada por meio de ferramentas desenvolvidas especificamente para esta finalidade, como, por exemplo, o pacote irace [López-Ibáñez et al. 2011], que automatiza o processo de seleção de parâmetros buscando a melhor configuração a partir de um conjunto de instâncias informado para o problema.

Por fim, é levantada a possibilidade de criar uma “máquina” que seja capaz de criptografar e descriptografar mensagens com base nas chaves geradas pelo algoritmo desenvolvido, para observar o seu funcionamento em situações próximas das reais.

Referências

- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549.
- Government, U. (2011). *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. General Books.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *SCIENCE*, 220(4598):671–680.
- López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., and Birattari, M. (2011). The irace package, iterated race for automatic algorithm configuration. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium.
- Miller, F. (1882). *Telegraphic Code to Insure Privacy and Secrecy in the Transmission of Telegrams*. C.M. Cornwell.
- Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656.
- Tomassini, M. and Perrenou, M. (2001). Cryptography with cellular automata. In *Applied Soft Computing I*, pages 151–160.
- Villela, S. M. and Carvalho, L. A. V. (2007). O resfriamento simulado no projeto ótimo de autômatos celulares para a geração de chaves em criptografia de fluxo. In *ENIA – VI Encontro Nacional de Inteligência Artificial*, pages 1082–1091.
- Wolfram, S. (1982). Cellular automata as simple self-organizing systems. *S. Caltech Preprint CALT-68-938*.
- Wolfram, S. (1985). Cryptography with cellular automata. In *Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA*, pages 429–432.
- Wolfram, S. (2002). *A new kind of science*. Wolfram-Media.
- Zeng, K., Yang, C. H., Wei, D. Y., and Rao, T. R. N. (1991). Pseudorandom bit generators in stream-cipher cryptography. *Computer*, 24(2):8–17.

Predição de Desempenho de Alunos no Ensino a Distância via Mineração de Processos

Fabio Q. Tanaka, Gabriel Lucas C. da Silva, Sarajane M. Peres, Marcelo Fantinato

¹Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
São Paulo – SP – Brasil

{fabio.quiochi.tanaka, gabriel.lcsilva, sarajane, m.fantinato}@usp.br

Abstract. *This paper presents results of the application of the Multilayer Perceptron artificial neural network to predict the performance of students in a distance learning program. We used a multiclass prediction model, with data from the interaction process of students in the virtual learning environment. This type of unstructured process has been not properly addressed in process mining since it is vaguely defined and characterized by unpredictability of external factors, which makes it difficult to discover patterns that describe its behavior.*

Resumo. *Este artigo apresenta resultados da aplicação da rede neural artificial Multilayer Perceptron para prever o desempenho de alunos em um curso a distância. Um modelo de predição multiclasse foi usado, com dados do processo de interação dos alunos no ambiente virtual de aprendizagem. Trata-se de um processo não estruturado (chamado de “espaguete”), um tipo de processo ainda não tratado adequadamente em mineração de processos, por ser vagamente definido e caracterizado pela imprevisibilidade de fatores externos, o que dificulta a descoberta de padrões que descrevam seu comportamento.*

1. Introdução

O ensino a distância apresenta novas oportunidades incluindo a análise computacional do comportamento de alunos no Ambiente Virtual de Aprendizagem (AVA) para apoiar a tomada de decisão [Do Carmo et al. 2012]. Um exemplo é prever o desempenho de alunos em disciplinas, em termos de notas, dependendo do processo de interação seguido por eles. Há linhas de pesquisa que defendem AVAs com navegação flexível, permitindo aos alunos estabelecerem suas próprias estratégias de estudo junto aos objetos de aprendizagem, flexibilizando o processo de aprendizado visando maior eficácia [Chatzisawa et al. 2010, Bacon and MacKinnon 2014]. Contudo, é possível que esse excesso de flexibilidade leve alguns alunos ao resultado contrário, como um efeito colateral.

Diferentes aspectos interdisciplinares colaboram para esse contexto. A Gestão de Processos de Negócio (BPM – *Business Process Management*) organiza a modelagem, execução e monitoramento de processos executados por um conjunto de indivíduos em busca de um objetivo comum [Aalst et al. 2003]. A mineração de dados extrai padrões escondidos ou conhecimento potencialmente útil a partir de um grande volume de dados, com o objetivo de auxiliar o processo de tomada de decisão [Han et al. 2011]. Ambas áreas usadas em conjunto têm sido chamadas de mineração de processos [Aalst 2011], para a qual os dados minerados são representados por *logs* de execução do processo.

Processos apoiados por AVAs representam um tipo de processo ainda não tratado adequadamente em mineração de processos: os processos não estruturados, chamados de processos “espaguete” [Aalst 2011]. Diferentemente de processos estruturados (os processos “lasanha”), processos “espaguete” são vagamente definidos e caracterizados pela imprevisibilidade de fatores externos, o que dificulta a descoberta de padrões que descrevam seu comportamento [Aalst 2011]. Um estudo prévio indicou que muitas técnicas amplamente aplicadas em mineração de dados têm sido negligenciadas em mineração de processos, principalmente de inteligência computacional e aprendizado de máquina [Maita 2015]. Um exemplo característico são as redes neurais artificiais e as máquinas de vetores suporte, muito usadas em diferentes tarefas de mineração de dados [Wu et al. 2007], e pouco usadas em mineração de processos [Maita et al. 2015].

Resultados preliminares foram apresentados em [Maita et al. 2017] considerando a aplicação de redes neurais *Multilayer Perceptron* (MLP) nesse contexto, para prever o desempenho de alunos, em um cenário real, com um modelo de classificação binário. Já o estudo apresentado neste artigo evolui a experimentação para um modelo multiclasse. Outros refinamentos realizados são apresentados neste artigo.

Este artigo apresenta: fundamentação teórica (Seção 2), análise de trabalhos relacionados (Seção 3), apresentação do problema de classificação para descoberta e análise de processos e consequente previsão de desempenho (Seção 4), descrição do conjunto de dados usado e pré-processamento realizado (Seção 5), projeto do experimento e análise de resultados (Seção 6) e considerações finais (Seção 7).

2. Fundamentação Teórica

Nesta seção, são apresentados brevemente os principais conceitos teóricos abordados neste artigo: BPM, mineração de dados, mineração de processos e redes neurais MLP.

2.1. Gestão de Processos de Negócio

Um processo de negócio é um conjunto de tarefas inter-relacionadas que devem ser seguidas para que metas organizacionais sejam alcançadas [Weske 2012, Lindsay et al. 2003]. Gestão de Processos de Negócio (BPM – *Business Process Management*) se refere ao uso de técnicas, métodos, ferramentas, especificações, linguagens, *middlewares*, *frameworks*, e arquitetura de modelos de software e serviços para apoiar as atividades do ciclo de vida de processos [Weske 2012, Dumas et al. 2013]. Diferentes tipos de organização podem ser apoiados por BPM, incluindo organizações educacionais como tratadas neste artigo.

BPM é primordial para uma bem sucedida gestão de tecnologia da informação alinhada às demandas estratégicas de uma organização [Aalst et al. 2003, Jensen et al. 2008]. Para auxiliar uma organização em obter vantagem competitiva, o ciclo de vida de BPM pode incluir: modelagem, implementação, execução, monitoramento, auditoria e melhoria de processos [Weske 2012, Dumas et al. 2013]. Durante as fases de análise e otimização, os *logs* de execução e de monitoramento podem ser avaliados para encontrar potenciais sugestões de melhoria nos processos, levando ao recomeço do ciclo de vida para um novo estágio de modelagem [Weske 2012].

2.2. Mineração de Dados

Mineração de dados é usada para extrair padrões escondidos ou conhecimento potencialmente útil a partir de um grande volume de dados, com o objetivo de auxiliar o processo

de tomada de decisão [Han et al. 2011]. De maneira geral, organizações têm sido efetivas na extração e armazenamento de dados. Entretanto, transformar dados em conhecimento útil ainda é um problema complexo. Mineração de dados explora e analisa tais dados, automaticamente ou semiautomaticamente, para descobrir padrões relevantes, ou conhecimento útil e inesperado, escondidos no conjunto de dados.

As tarefas primárias de mineração são: predição categórica (ou classificação); predição numérica (tipo “regressão” e tipo “identificação de tendências”); análise de agrupamentos; descoberta de padrões frequentes, regras de associação e correlação; análise de “outliers” e ruído; análise de evolução; e análise de dados estruturados. [Han et al. 2011]. Para cada tarefa de mineração de dados, várias técnicas foram propostas [Han et al. 2011, Wu et al. 2007], incluindo técnicas de inteligência computacional [Wang and Fu 2005, IEEE 2008] e de aprendizado de máquina [Mitchell 1997].

Inteligência computacional refere-se ao desenvolvimento de software inteligente baseado em aprendizado, adaptação ou evolução [IEEE 2008]. Aprendizado de máquina dedica-se estritamente à indução de hipótese baseada em experiência, compreendendo principalmente técnicas não cobertas pela inteligência computacional. Ambas têm se mostrado eficientes em resolver tarefas de mineração de dados por lidarem apropriadamente com dados de contexto mal definidos, para os quais modelos exatos são inviáveis, o que é muito comum em dados de problemas do mundo real [Abonyí et al. 2005]. Alguns exemplos de tais técnicas são aquelas de raciocínio aproximado (como lógica Fuzzy e raciocínio probabilístico), raciocínio evolutivo (como técnicas de computação evolutiva) e raciocínio indutivo (como redes neurais e máquinas de vetores suporte).

2.3. Mineração de Processos

Mineração de processos é uma área de pesquisa emergente que se encontra entre BPM e mineração de dados [Aalst 2011]. O objetivo dessa mineração é percorrer *logs* de eventos para extrair conhecimento que permita a descoberta, o monitoramento e a melhoria de processos. Busca-se extrair conhecimento sobre eventos de execução de processos por meio da descoberta de associações entre variáveis e padrões de comportamento [Aalst 2011]. Embora a mineração de processos tenha sido introduzida há cerca de 20 anos, existem tipos de processos para os quais ainda não foram propostas técnicas de mineração satisfatórias, como os processos “espaguete” (não estruturados). Diferentemente de processos “lasanha” (estruturados), processos “espaguete” são vagamente definidos, dificultando a descoberta de padrões que descrevam seu comportamento [Aalst 2011].

Existem diferentes tipos de mineração de processos dependendo do tipo de conhecimento que se deseja descobrir em termos de padrões relevantes a partir dos *logs* de eventos analisados. Os principais tipos de mineração de processos são: descoberta de processos, que produz um modelo de processo ainda desconhecido baseado somente no *log* de eventos; conformidade de processos, que compara um modelo de processo já conhecido com um *log* de eventos do mesmo processo para verificar se o modelo está de acordo com a realidade; e melhoria de processo, que auxilia na modificação de um modelo de processo já existente baseado no *log* de eventos de execução do processo [Aalst 2011].

2.4. Redes Neurais Artificiais MLP

Um *perceptron* (neurônio) pode ser formalmente definido via equação 1, em que: w é o vetor de pesos aplicado nas entradas do neurônio, x é o vetor de entrada, b é um vetor de

tendência que pode ser positivo ou negativo e é ajustado a cada ciclo de aprendizagem juntamente com os vetores de pesos, φ é a função de ativação responsável por uma resposta do neurônio após uma entrada, e n é a dimensão de entrada fornecida ao neurônio.

$$y = \varphi \left(\sum_{i=1}^n w_i x_i + b \right) = \varphi \left(\mathbf{w}^T \mathbf{x} + \mathbf{b} \right), \quad (1)$$

Redes Multilayer Perceptron (MLP) são formadas por *perceptrons* interconectados organizados como: uma camada de entrada, que recebe a informação a ser processada pelos neurônios; uma ou mais camadas ocultas, que são os grupos de neurônios entre as camadas de entrada e de saída; e a camada de saída, que produz a resposta. Em redes MLP, a informação é propagada progressivamente. Cada neurônio está conectado a todos os outros da camada seguinte, sem retroalimentação. Um algoritmo de aprendizagem deve ajustar os pesos das conexões dos neurônios e, assim, extrair as características de um problema, atuando usualmente de duas formas: sequencial, com erros analisados em cada iteração, ou em lote, com erros analisados ao final de uma época (após todos os dados de treinamento serem processados pela rede).

O treinamento via retropropagação do erro é um dos mais usados em redes neurais de aprendizagem supervisionada. Sua proposição [Werbos 1974] permitiu um importante avanço nessa área [Haykin 2009]. Na retropropagação, um erro é determinado em uma iteração m , avaliada ao subtrair o valor resultante da iteração $y_j(m)$ do valor desejado $d_j(m)$. O erro quadrático médio de uma iteração s , representado por ϵ_{mean} , é responsável por representar uma função de custo (Eq. 2). Essa função de custo representa a necessidade de ajustar os parâmetros da rede neural para obter a aprendizagem (aproximação) de um padrão. Nesse ajuste, são aplicadas correções Δw_{ji} nos pesos sinápticos, em que i é o número de valores de entrada em um neurônio e j é o identificador do neurônio. A correção é proporcional à derivada parcial da soma dos erros na última camada $\epsilon(m)$ sobre Δw_{ji} , isto é, $\partial \epsilon(m) / \Delta w_{ji}$, que define a taxa a ser aplicada ao peso Δw_{ji} .

$$\epsilon_{mean} = \frac{1}{M} \sum_{m=1}^M \epsilon(m) \quad (2)$$

3. Trabalhos Relacionados

Poucos trabalhos têm aplicado redes neurais no contexto de mineração de processos. Apenas duas arquiteturas de redes neurais foram aplicadas: (i) *feedforward multilayer*, que implementa aprendizado supervisionado para resolver a tarefa de predição categórica em dois trabalhos [Ho et al. 2004, Thomas et al. 2006] e a tarefa de predição numérica (do tipo “regressão”) em três trabalhos [Ho et al. 2006, Tsang et al. 2006, Zhang et al. 2011]; e (ii) *competitive layer*, que implementa aprendizado não supervisionado para resolver a tarefa de agrupamento em dois trabalhos [Song et al. 2013, Seret et al. 2014]. Os tipos de mineração de processo implementados nesses artigos foram: descoberta [Song et al. 2013, Seret et al. 2014], conformidade [Ho et al. 2006] e melhoria [Ho et al. 2006, Tsang et al. 2006, Thomas et al. 2006, Zhang et al. 2011].

Descoberta de processos foi alvo de estudo com uma arquitetura de rede *Self Organizing Maps* (SOM) [Song et al. 2013, Seret et al. 2014]. No primeiro caso, os auto-

res resolvem uma tarefa de agrupamento para melhorar o entendimento de modelos de processo descobertos a partir da análise dos eventos dos processos de um hospital. No segundo, os autores combinaram SOM com o algoritmo *k-means* para analisar o comportamento de dados de grupos de clientes, descobrindo o processo relacionado às ações dos clientes com o passar do tempo. Conformidade de processos foi implementada combinando redes neurais e sistemas de processamento analítico *online* (*On-Line Analytic Processing* – OLAP) [Ho et al. 2004], abordando a simplificação do fluxo de informação em um processo logístico e a identificação de anomalias comportamentais durante a execução do processo. Para ilustrar sua abordagem, os autores estudaram uma companhia de aviação, para a qual o processo logístico é altamente suscetível a riscos envolvendo operações de aeronaves. Quatro artigos tratam melhoria de processos. No primeiro [Thomas et al. 2006], uma rede neural supervisionada é usada como um otimizador de regras Fuzzy, para construir um sistema capaz de modelar um processo da melhor maneira possível. Nos outros dois trabalhos [Ho et al. 2006, Tsang et al. 2006], redes neurais supervisionadas foram usadas como um preditor da qualidade do fluxo de trabalho e para tratar melhorias para o processo em tempo real. No quarto trabalho [Zhang et al. 2011], uma rede neural supervisionada é usada para melhorar um processo via otimização de parâmetros, como proposta para que operadores humanos ajustem o processo.

Em comparação com esses trabalhos relacionados, tanto o trabalho previamente apresentado [Maita et al. 2017] quanto o trabalho apresentado neste artigo aplicam uma arquitetura de rede neural MLP, implementando aprendizado supervisionado, para resolver a tarefa de predição categórica; no primeiro caso, uma predição binária, e, no segundo caso, uma predição multiclasse. Em termos de tipo de mineração de processos, também ambos os trabalhos realizam descoberta de processo, em que um modelo de processo é descoberto para ser usado na predição do comportamento dos alunos.

4. Apresentação do Problema

O objetivo deste estudo é explorar redes neurais na análise de processos “espaguete” provenientes de um contexto de ensino a distância. Para isso, dados de navegação de alunos sobre um AVA¹ de um curso de especialização semipresencial foram tomados como representação do processo que modela o comportamento desses alunos e foram enriquecidos com dados relacionais para apoiar a modelagem de um problema de classificação categórica. A classificação pretendida refere-se à predição do desempenho do aluno no curso de especialização em relação a nota desses alunos em disciplinas do curso.

A arquitetura da predição proposta está resumida na Figura 1. O objetivo é modelar, via uma rede neural MLP, o comportamento padrão de estudo dos alunos já egressos do curso, relacionando esse comportamento a classes de desempenho (“desistência”, “insatisfatório”, “regular”, “destaque”). Assim, uma vez que a rede neural construiu o mapeamento, ela pode ser usada para prever o desempenho de novos alunos. A informação sobre essa predição fornece aos alunos, professores e coordenadores do curso, um conhecimento estratégico para tomada de decisão dentro do processo de ensino e aprendizagem.

O processo seguido pelos alunos no AVA pode ser classificado como processo do tipo “espaguete”, por ser vagamente definido e caracterizado pela imprevisibilidade

¹Detalhes sobre o sistema AVA referente ao contexto dos dados sob análise neste estudo podem ser encontrados em [Junior et al. 2013, Peres et al. 2012].

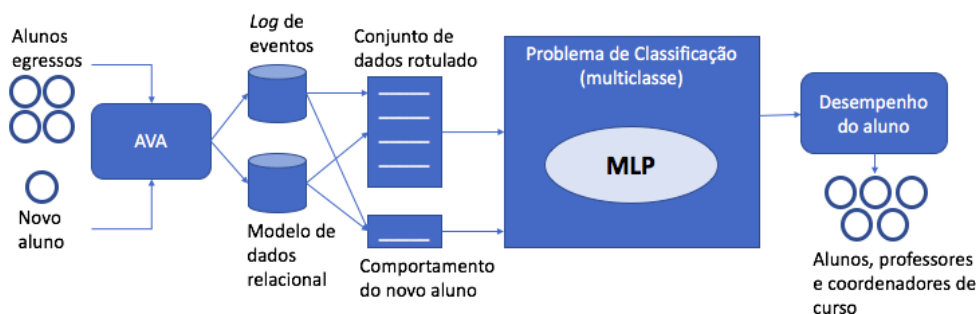


Figura 1. Arquitetura proposta para predição do desempenho do aluno

de fatores externos, incluindo as ações dos próprios alunos. Os alunos conseguem, por exemplo, acessar os conteúdos que desejam e na ordem que desejam, não sendo obrigados a seguir qualquer tipo de estrutura pré-determinada. Processos “espaguete” são mais difíceis de tratar, devido à grande variabilidade de instâncias possíveis derivadas de um mesmo conjunto de atividades (incluindo, por exemplo, mudança da ordem de execução e realização de laços). A modelagem de problemas dessa natureza pode resultar em um conjunto de dados com alta esparsidade, podendo chegar a até 99% de esparsidade, como o visto no trabalho de [Maita 2015]. Essas características dificultam a descoberta de padrões que descrevam o comportamento de processos dessa natureza.

5. Conjunto de dados e pré-processamento

A plataforma Moodle possui uma funcionalidade que permite criar arquivos de *log* referentes a todas as ações de navegação executadas pelos usuários dos sistemas construídos sobre ela. Esses arquivos de *log* permitiram construir o *log* de eventos usado como base para a descoberta de processos neste trabalho. A Figura 2 ilustra um trecho do *log* de eventos com as informações básicas² usadas nos experimentos realizados. Cada linha do *log* de eventos diz respeito a uma ação (“Ação”) de um aluno, dentro de uma seção do curso na plataforma (“Curso”), diante de um objeto de aprendizagem (“Informação”), executada em um momento do tempo (“Hora”).

3	Curso	Hora	endereço IP	Nome completo	Ação	Informação
144	Módulo IV	2013 julho 19 15:32			course view	Profissão docente e educação especial/inclusiva
145	Módulo IV	2013 julho 19 15:32			course view	Profissão docente e educação especial/inclusiva
146	EVC	2013 julho 19 15:31			user login	1009
147	EVC	2013 julho 19 15:31			course view	Ética, Valores e Cidadania na Escola
148	REPIV	2013 julho 19 12:21			resource view	Legenda - VA16
149	Educação especial/inclusiva	2013 julho 19 12:18			course view	Educação especial/inclusiva
150	Módulo IV	2013 julho 19 12:18			course view	Profissão docente e educação especial/inclusiva

Figura 2. Trecho do *log* de eventos

Visto que o problema sob resolução trata da descoberta e análise do processo seguido por cada um dos alunos na plataforma, o *log* de eventos foi transformado em um conjunto de dados que descreve o comportamento do aluno. Nesse conjunto de dados, cada aluno é descrito por meio da contagem de vezes que ele realizou uma “ação-atividade”, isto é, os campos “Ação” e “Atividade” formaram a base de criação dos atributos descritivos do processo executado pelo aluno. No processo de criação do conjunto

²Os campos “endereço IP” e “Nome completo” foram suprimidos da figura para manter as condições de anonimato dos usuários.

de dados, anomalias presentes no *log* de eventos original (ou seja, registros referentes a usuários de teste de sistema, registros com informações incompletas por conta de instabilidades da plataforma, e registros referentes a erros de sistema) foram identificadas e excluídas. A fim de diminuir o problema de alta dimensionalidade na matriz de dados, atividades para as quais a contagem total de ocorrências para todos os alunos foi menor do que dez foram arbitrariamente excluídas. Essa exclusão resultou na retirada de 173 dimensões, das 645 originais, do conjunto de dados.

A proposta de mineração de processos foi modelada como uma tarefa de classificação. Assim, foi necessário classificar os alunos de acordo com seu desempenho no curso. O desempenho foi dividido em quatro classes: 0 (desistentes) – para alunos que desistiram do curso (com 108 alunos); 1 (insatisfatório) – nota final $< 7,0$ (com 76 alunos); 2 (regular) – nota final entre $7,0$ e $8,5$ (com 38 alunos); 3 (destaque) – nota final $\geq 8,5$ (com 667 alunos). A informação de nota final não decorre do *log* de eventos, sendo obtida a partir do modelo de dados relacional do AVA. Essa informação também apresentava inconsistências, as quais resultaram na exclusão da análise dos processos de alguns alunos nesta experimentação. Devido ao desbalanceamento existente em relação às classes no conjunto de dados, foi realizada uma amostragem aleatória de 38 registros (alunos) para cada classe. Esse conjunto de dados amostrado está nomeado como *conjunto de dados A* para fins da descrição do experimento e análise de resultados apresentados neste artigo. Um trecho do conjunto de dados, que possui 472 atributos, é apresentado na Figura 3. Nessa figura, um aluno é representado pelo número de vezes que ele acessa o sistema e os diferentes tipos de objetos de aprendizagem: o segundo aluno na figura, por exemplo, acessou o sistema 237 vezes e o curso do “*Prof. docente ...*” sete vezes.

user login	user logout	resource view - Legenda	course view - Prof. docente ...	classe
155	177	0		0	2
237	8	3		7	3
237	20	1		43	3
134	41	0		0	3
291	11	23		3	3
266	16	25		123	3
35	1	1		0	0
60	7	0		0	1

Figura 3. Trecho do conjunto de dados A

Embora já diferente do original, o conjunto de dados criado ainda possui alta dimensionalidade. A fim de reduzir essa dimensionalidade e contribuir para a construção de um espaço de busca mais simples, no qual hipóteses de menor complexidade pudessem ser eficientemente exploradas [Haykin 2009], foi realizado um procedimento de seleção de atributos sobre o *conjunto de dados A*. Esse procedimento combinou³:

- Análise de componentes principais [Han et al. 2011]: para estimar o número mínimo de dimensões necessárias para alcançar uma boa representatividade dos dados. Como resultado, foi observado que com dez dimensões (as dez primeiras componentes principais) seria possível representar 93,3% da informação contida no conjunto de dados. A Figura 4 mostra a progressão da variância presente em

³Para implementação desses procedimentos, foram usadas as bibliotecas *sklearn.decomposition.PCA* e *sklearn.feature-selection.SelectKBest* [Pedregosa et al. 2011].

cada dimensão do conjunto de dados. Percebe-se que a incorporação de mais dimensões não traria um aumento significativo na representatividade dos dados.

- Teste estatístico χ^2 [Liu and Setiono 1995]: análise da distribuição de classes em relação à variabilidade dos valores assumidos por cada um dos atributos do conjunto de dados, medindo a correlação entre atributos descritivos e atributo de classe. A partir do conhecimento gerado pelo PCA, foi decidido manter os dez atributos mais relevantes, de acordo com esse teste estatístico, na formação de um novo conjunto de dados nomeado aqui de *conjunto de dados B*.

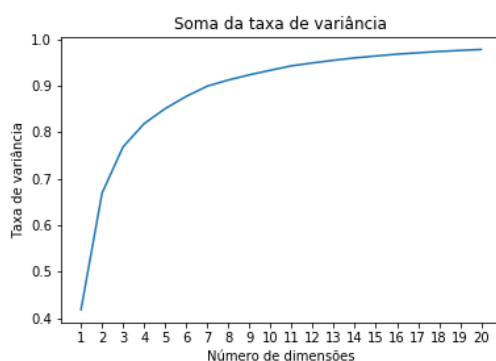


Figura 4. Progressão da taxa de variância dos atributos (dimensões) do conjunto de dados A. A partir da vigésima dimensão, a variância é mínima e foi suprimida do gráfico para fins de melhoria na visualização.

Para apresentação à rede neural MLP, os conjuntos de dados foram normalizados no intervalo $[-1, 1]$, por meio de uma normalização por *z-score* [Han et al. 2011].

6. Experimentos e resultados

Foram desenvolvidos modelos de rede neural MLP para ambos os conjuntos de dados (*A* e *B*). Todos os modelos foram criados sob uma estratégia de teste *holdout*, considerando uma partição de 70% dos dados para treino e 30% dos dados para teste. Os modelos criados foram avaliados no conjunto de teste usando o erro quadrático médio (erro total cometido pela rede neural), acurácia por classe e acurácia total.

A arquitetura da MLP, usada para a criação de todos os modelos, contou com uma camada de neurônios escondidos e uma camada de saída com quatro neurônios, sendo que cada um dos neurônios teve o papel de representar uma das classes do problema. Assim, uma saída da rede na forma $\langle 1 \ 0 \ 0 \ 0 \rangle$ indica que o dado sob análise é classificado como pertencente à classe 0, uma saída na forma $\langle 0 \ 1 \ 0 \ 0 \rangle$ indica que o dado pertence à classe 1, e assim por diante. Todos os pesos e *bias* da arquitetura foram inicializados aleatoriamente no intervalo $[-0, 5, 0, 5]$. Como condição de parada, duas situações foram assumidas: número máximo de épocas (500) ou manutenção do erro quadrático médio mínimo $\leq 0,01$ por 50 épocas, o que ocorrer primeiro⁴.

Para a calibração dos meta-parâmetros da MLP, foi usada uma estratégia de combinação em grade, variando os parâmetros de acordo com o especificado na Tabela 1.

⁴A criação dos modelos foi realizada com apoio da ferramenta Keras [Chollet et al. 2015] e da biblioteca TensorFlow [Abadi, M. et al. 2015].

Como resultado da combinação de todos os valores para os parâmetros, foram obtidos 1.728 modelos classificadores para cada um dos conjuntos de dados (*A* e *B*).

Tabela 1. Meta parâmetros da MLP e intervalos de variação na busca em grade

Parâmetro	Descrição	Valores aplicados
# de neurônios na camada escondida (neur.)	Realiza uma extração de características dos dados de entrada. O número de neurônios define a dimensionalidade do espaço de características criado.	[150; 200; 250; 300] para o conjunto de dados <i>A</i> ; [10; 20; 35; 50] para o conjunto de dados <i>B</i> .
Taxa de aprendizagem inicial (α inicial)	Parâmetro que regulariza as alterações de pesos.	[0,1; 0,5; 1]
Fator de decaimento linear (em função da época de aprendizado) (decay)	Fator de controle da taxa de aprendizado, permitindo seu decaimento durante o aprendizado.	[0; 0,01]
Momentum η	Parâmetro que permite a construção de uma memória na alteração de pesos. Útil para evitar mínimos locais.	[0,1; 0,5; 1]
Nesterov momentum (Nest.)	Estratégia de regularização do momentum.	[sim; não]
Função de ativação da camada escondida (Φ_{ce})	Define a intensidade de ativação do neurônio da camada escondida.	[tangente hiperbólica; sigmóide]
Função de ativação da camada de saída (Φ_{cs})	Define a intensidade de ativação do neurônio da camada de saída.	[tangente hiperbólica; sigmóide; softmax]
Tipo de treinamento	Número de dados que são processados antes da mudança dos pesos.	[1 - sequencial; todos – em lote]

O desempenho dos 1.728 modelos, para cada conjunto de dados, foi aferido pela análise do comportamento do erro total da rede neural no conjunto de teste. Nela, 106 alunos foram usados para treinamento do modelo e 46 alunos restantes foram usados para teste. Os meta-parâmetros dos oito melhores modelos para o conjunto de dados *A* e dos 25 melhores modelos para o conjunto de dados *B*, obtidos nessa estratégia, foram usados em um novo esforço de construção de modelos. Nessa nova estratégia, foi aplicada validação cruzada com 5 *folds*, envolvendo 100% dos dados. O desempenho dos modelos obtidos nessa etapa foi aferido via acurácia média (e desvio padrão) na validação cruzada. Os três melhores modelos para cada um dos conjuntos de dados são apresentados na Tabela 2.

A partir dos resultados obtidos, é clara a dificuldade dos classificadores em lidar com o conjunto de dados de alta dimensionalidade. Nesses casos, a melhor acurácia foi de 65% de classificações corretas. Com a seleção de atributos, os classificadores melhoraram seus desempenhos. No melhor resultado obtido, foi realizado um estudo sobre o tipo de erro cometido pelo classificador. Para isso, a matriz de confusão gerada no teste do classificador de melhor desempenho na validação cruzada foi analisada e mostrou que os erros são, na maioria das vezes, referentes à classificação do aluno em uma categoria de desempenho abaixo da sua. Esse padrão não ocorreu em apenas 14% das classificações erradas. Ou seja, em um processo de aconselhamento, *seria dito a um aluno bom que ele precisa estudar mais* (o erro do classificador não traria uma consequência ruim).

Tabela 2. Melhores classificadores obtidos para cada conjunto de dados

<i>Conjunto de dados A</i>								
# neur.	α inicial	decay	η	Nest.	Φ_{ce}	Φ_{cs}	acurácia média	σ
300	0,1	0	1	FALSE	tanh	sigmoid	0,65	0,001
200	0,5	0	0,1	TRUE	sigmoid	sigmoid	0,60	0,010
250	0,5	0	0,5	TRUE	sigmoid	sigmoid	0,58	0,010
<i>Conjunto de dados B</i>								
# neur.	α inicial	decay	η	Nest.	Φ_{ce}	Φ_{cs}	acurácia média	σ
35	1	0	0,1	TRUE	tanh	sigmoid	0,70	0,0006
50	0,5	0,01	1	TRUE	sigmoid	softmax	0,71	0,0005
10	1	0	0,1	TRUE	tanh	sigmoid	0,71	0,0003

7. Conclusão

Neste artigo, foram apresentados os resultados da aplicação da rede neural MLP para predição do desempenho de alunos em um curso a distância. Os alunos foram representados em termos de seu comportamento dentro de um AVA, ou seja, representados a partir das características do processo não estruturado que seguem ao navegar pela plataforma, caracterizando uma forma indireta de descoberta de processo. Essa iniciativa de predição é motivada pela possibilidade de monitorar o comportamento de alunos que passem pelo mesmo processo de aprendizado (mesmo curso apoiado pela mesma plataforma AVA) para aconselhá-los quando seu comportamento indicar baixo desempenho.

Os resultados obtidos, principalmente considerando que a amostra de dados usada nos experimentos é pequena, indicam que a abordagem de mineração de processos aplicada apresenta razoabilidade, estimulando a continuidade dos esforços para alcançar resultados mais significativos. O tamanho reduzido da amostra foi amenizado pelo trabalho de seleção de atributos. Analisando o melhor classificador obtido, observa-se resultados de 78% de acertos, e erros localizados em situações de baixo risco, ou seja, erros do tipo falsos positivos (18%), indicando que comportamento do aluno está sendo subestimado. Nesse caso, os alunos estarão sendo estimulados a melhorar seu comportamento. Ainda, essa análise de comportamento pode servir como um apoio para professores e coordenadores de curso que desejam evitar resultados ruins nos cursos que oferecem.

A exploração do problema discutido neste artigo ainda segue em evolução. Há dois estudos em andamento que exploram o problema sob diferentes aspectos:

- O primeiro estudo modela o problema sob um ponto de vista temporal. A base de dados está sendo enriquecida com dados de dois oferecimentos do mesmo curso⁵, de forma que seja possível criar classificadores baseados em módulos do curso. Assim, o comportamento do aluno em períodos específicos do curso pode ser analisado e o aconselhamento pode, de fato, ser feito em tempo de gerar mudança de comportamento, ou seja, ser feito de forma preemptiva. Além disso, o desbalançamento de dados deverá ser minimizado com uso de amostras ponderadas.
- O segundo estudo tem o objetivo de apresentar o modelo formal do processo que descreve o comportamento do aluno na plataforma AVA. O processo sob estudo

⁵O curso foi oferecido três vezes, o que permite um ambiente de treinamento e teste distribuído no tempo.

é do tipo “espaguete”, e muito pouco tem sido desenvolvido para formalizar o comportamento deste tipo de processo. No referido estudo, está sendo modelado um algoritmo genético para descoberta de um modelo de processo a partir do *log* de eventos. Se a descoberta for possível, os coordenadores de curso terão acesso a um conhecimento que os permitirá trabalhar em prol da melhoria das condições de aprendizado dos alunos, buscando alterar, de forma natural, o comportamento da classe de alunos que não conseguem bons resultados no curso.

8. Agradecimentos

Os autores agradecem o apoio do Programa de Educação Tutorial (SESu/MEC).

Referências

- Aalst, W. M. P. (2011). *Process Mining – Discovery, Conformance and Enhancement of Business Processes*. Springer, 1st edition.
- Aalst, W. M. P., Hofstede, A. H. M., and Weske, M. (2003). Business process management: A survey. In *Proc. of the Int. Conf. on Bus. Proc. Man.*, pages 1–12. Springer.
- Abadi, M. et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. <http://tensorflow.org/>.
- Abonyí et al., J. (2005). Computational intelligence in data mining. *Informatica*, 29(1):3–12.
- Bacon, L. and MacKinnon, L. (2014). A flexible framework for metacognitive modelling and development. In *Int. Conf. on e-Learning*, pages 7–14.
- Chatzisawa, S. et al. (2010). A dynamic environment for distance learning. In *Int. Conf. on Computer Supported Education*, volume 1, pages 398–401.
- Chollet, F. et al. (2015). Keras. <https://github.com/fchollet/keras>.
- Do Carmo, A. et al. (2012). Information visualization to support decision making in the context of distance learning: A study with VLE TelEduc. In *38th Latin America Conf. on Informatics*.
- Dumas et al., M. (2013). *Fundamentals of Bus. Process Management*. Springer.
- Han et al., J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3rd edition.
- Haykin, S. (2009). *Neural networks and learning machine*. Prentice Hall, 3 edition.
- Ho, G. et al. (2004). An intelligent information infrastructure to support the streamlining of integrated logistics workflow. *Exp. Sys.*, 21(3):123–137.
- Ho et al., G. T. S. (2006). An intelligent production workflow mining system for continual quality enhancement. *Int. J. of Adv. Manufac. Tech.*, 28(7):792–809.
- IEEE (2008). Computational Intelligence. <http://ewh.ieee.org/r7/montreal/cis/>.
- Jensen et al., C. T. (2008). Leveraging SOA, BPM and EA for strategic business and IT alignment. *IBM Bus. Proc. Man. J.*, 1:10.1–10.28.
- Junior, M. P. S. et al. (2013). Experimenting with design thinking in requirements refinement for a learning management system. *Simp. Bras. de Sist. de Inf.*, pages 182–193.

- Lindsay et al., A. (2003). Business processes-attempts to find a definition. *Inf. and Sof. Tech.*, 45(15):1015–1019.
- Liu, H. and Setiono, R. (1995). Chi2: Feature selection and discretization of numeric attributes. In *IEEE Int. Conf. on Tools with Artif. Intel.*, pages 388–391.
- Maita, A. R. C. (2015). Um estudo da aplicação de técnicas de inteligência computacional e de aprendizado de máquina em mineração de processos de negócio. Master's thesis, Universidade de São Paulo.
- Maita, A. R. C. et al. (2015). Process mining through artificial neural networks and support vector machines: A systematic literature review. *Bus. Proc. Man. J.*, 21(6):1391–1415.
- Maita, A. R. C. et al. (2017). Mining unstructured processes: An exploratory study on a distance learning domain. In *Int. J. Conf. on Neural Networks*, pages 3240–3247, Anchorage, AK, USA.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw Hill, New York, NY, USA, 1st edition.
- Pedregosa, F. et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Peres, S. M., Fantinato, M., Araújo, U., Soledade Junior, M. P., Freitas, R. S., Azevedo, F. M., and Barbosa, H. M. A. (2012). Apoio ao aprendizado semipresencial: Uma aplicação no curso de especialização Ética, valores e saúde na escola. In *Congresso Brasileiro de Informática na Educação*, pages 1–10.
- Seret et al., A. (2014). A dynamic understanding of customer behavior processes based on clustering and sequence mining. *Exp. Sys.*, 41(10):4648–4657.
- Song et al., M. (2013). A comparative study of dimensionality reduction techniques to enhance trace clustering performances. *Exp. Sys.*, 40(9):3722–3737.
- Thomas et al., O. (2006). A fuzzy paradigm approach for business process intelligence. In *IEEE Int. Conf. on E-Com. Tech. and 3rd IEEE Int. Conf. on Ent. Comp., E-Com. and E-Serv.*, pages 26–29, Palo Alto. IEEE Comp. Soc.
- Tsang et al., K. F. (2006). Development of a distributed process mining system for reactive ion etching enhancement. In *IEEE Int. Conf. on Industrial Inf.*, pages 16–18, Singapore. IEEE Press.
- Wang, L. and Fu, X. (2005). *Data Mining with Computational Intelligence*. Springer, Berlin.
- Werbos, P. J. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. PhD thesis.
- Weske, M. (2012). *Business Process Management: Concepts, Languages, Architectures*. Springer, Berlin, 2nd edition.
- Wu, X. et al. (2007). Top 10 algorithms in data mining. *Knowl. Inform. Sys.*, 14(1):1–37.
- Zhang et al., J. L. (2011). Process optimization of candy production based on data mining. *Advanced Materials Research*, 282–283:662–665.

Um *pipeline* para predição de genes em dados metatranscriptômicos

Alessandra P. Alves de Oliveira², Ronnie C. O. Alves^{1,2}

¹Instituto Tecnológico Vale (ITV), Belém, Pará, Brazil

²Computer Science Graduate Program (PPGCC), Federal University of Pará (UFPA), Belém, Pará, Brazil

{alessandrapriscila.alves, alvesrco}@gmail.com

Abstract. *Through the study of metagenomic data it is possible to determine which organisms are present in a sample and what are their activities. There are gene prediction tools created to determine the organisms present in a sample, but there is the need for more tools to study the transcriptional data in order to understand the organism's behavior, such as TransGeneScan. This paper presents GeneFinder-TR, a pipeline for gene prediction in metatranscriptomic data. The pipeline's results were the best ones in half of the tests, obtaining up to 95.14% accuracy in its predictions.*

Resumo. *Através de estudos de dados metagenômicos, é possível definir quais organismos estão presentes em uma amostra e as suas atividades. Ferramentas para predição de genes em dados metagenômicos foram criadas para definir quais são os genes presentes na amostra, mas é necessário que mais ferramentas estudem os dados dos transcritos a fim de compreender o comportamento delas, como o TransGeneScan. Este artigo apresenta o GeneFinder-TR, um pipeline para predição de genes em dados metatranscriptômicos. O pipeline teve os melhores resultados em metade dos testes, obtendo até 95,14% de acurácia em suas predições.*

1. Introdução

Na bioinformática a predição de genes é o processo de identificação dos genes presentes em um genoma, ao determinar as posições de início ou fim de cada gene, chamados de códons, que são trincas de nucleotídeos. Esses códons formam combinações que definem os 20 tipos de aminoácidos. Os genes procariotos codificadores – que sintetizam proteína ou RNA – são formados por um códon de início, um número de códons variável após o primeiro e um códon de fim. Um dos fatores que dificulta esse processo é a existência de pequenas quantidades de DNA codificador de proteína ou de RNA em uma grande amostra de sequência de DNA [Stanke et al. 2006]. Essa dificuldade aumenta quando a predição de genes é realizada em amostras de comunidades microbiais, pois elas contêm múltiplos genomas que são analisados simultaneamente.

O conjunto de genomas de comunidades microbiais, denominado metagenoma, é coletado em diversos ambientes com água [Seshadri 2006], solo [Rondon et al. 2000] e animais [Frank e Pace 2008]. E os dados dessas amostras permitem definir, além dos tipos de organismos presentes, o comportamento e a interação deles entre si, sem a

necessidade de realizar a cultura dos mesmos [Ercolini 2013]. A metatranscriptômica é a área de estudo que avalia a atividade e as funções dos transcritos (RNA) das amostras ambientais.

Comunidades microbiais podem ter efeitos positivos e negativos em seus ambientes. Pesquisas extensas relacionam a um microbioma problemas de saúde como mal de Parkinson [Scheperjans et al. 2015], asma [Willner et al. 2012], diabetes tipo 2 [Qin et al. 2012] e doença de Crohn [Manichanh et al. 2006].

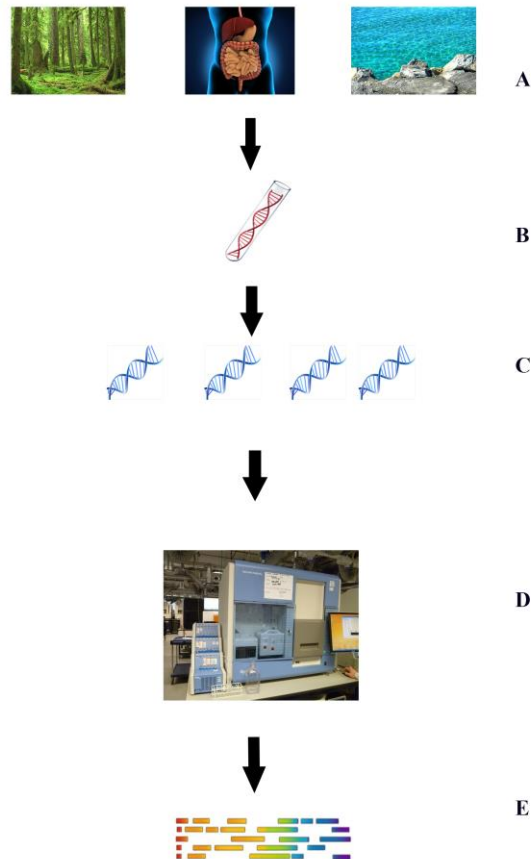


Figura 1. Passos para a predição de genes. Fonte: Autores.

O processo de predição de genes em um metagenoma necessita de que as amostras sejam digitalizadas por meio de sequenciamento (Figura 1). No item A, a amostra ambiental é colhida diretamente do microbioma, o item B mostra o isolamento do DNA. O item C mostra o processo de clonagem do DNA isolado que, no item D, passará pelo sequenciamento. O resultado do sequenciamento é demonstrado no item E. Os sequenciadores de segunda geração ou NGS sequenciam milhões de fragmentos ou sequências simultaneamente (*reads*) a um baixo custo e com rapidez, embora eles possam gerar erros de sequenciamento. Esses erros são divididos em (1) erros de substituição – no qual um nucleotídeo é trocado por outro –, (2) erros de eliminação – no qual um nucleotídeo é apagado da sequência –, (3) e erros de adição – no qual um nucleotídeo é adicionado a uma sequência. Quando esses erros ocorrem, as janelas de leitura (ORF – *Open Read Frame*) de um gene – as sequências compreendidas entre um códon de início e de fim (Figura 2) – sofrem uma mudança chamada *frameshift*. Algumas das ferramentas de predição de genes utilizam os códons como uma das

características para a predição de uma sequência de DNA codificador, o que pode afetar o resultado da predição.

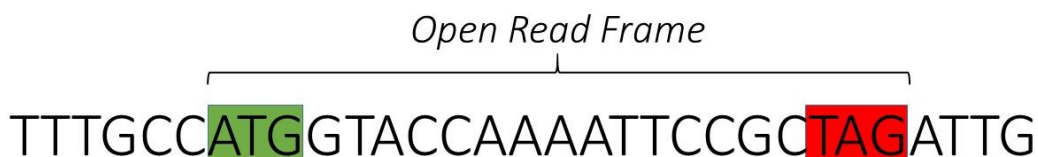


Figura 2. Exemplo de um *open read frame* em uma sequência. Fonte: Autores.

1.1. Trabalhos correlatos

O crescimento que a metagenômica apresentou desde o início dos anos 2000 foi acompanhado pela criação de diferentes ferramentas com o objetivo específico de realizar a predição de genes para dados colhidos de comunidades microbiais. As abordagens mais comuns encontradas para a predição de genes nessas ferramentas são (1) a abordagem por homologia, na qual uma sequência de DNA é comparada com outras sequências já verificadas no que é chamado de genoma de referência e a (2) abordagem *ab initio* que busca realizar a predição baseada em propriedades estatísticas e características próprias da sequência de DNA [Kunin et al. 2008].

Orphelia [Hoff et al. 2009] é uma ferramenta que realiza uma abordagem *ab initio*, utilizando redes neurais para analisar o conteúdo de GC (Guanina e Citosina) nas sequências. Ela tem dois modelos diferentes para as técnicas de sequenciamento mais comuns e identifica todos os ORFs presentes em uma sequência. Ela precisa que os *reads* tenham o tamanho mínimo de 60bp (pares de base ou nucleotídeos complementares). Ela também utiliza dados de códon (*códon usage*) e dicodons (seis nucleotídeos) para o processo de predição.

FragGeneScan [Rho 2010] é uma ferramenta baseada em um modelo oculto de Markov (HMM) que usa 7 superestados para a predição, utiliza dados de códons, módulos sequenciadores de erros e padrões para códons de início e de fim. Ela precisa que os *reads* tenham no mínimo 60bp e considera uma sequência como gene se ela tiver ao menos um códon de início ou de fim.

TransGeneScan [Ismail, Ye e Tang 2014] foi criada especificamente para trabalhar com sequências de dados metatranscriptômicos. Essa ferramenta usa um HMM baseado na da ferramenta FragGeneScan, considerando cada sequência como uma sequência de observação e faz o uso de regressão linear do conteúdo de GC.

MetaGene [Noguchi et al. 2006] faz uma estimativa entre as frequências de códon e de conteúdo de GC. Ela utiliza também frequências de dicodon, distribuição de frequências de ORFs e distância entre ORFs vizinhos.

1.2. Estudo de Caso

Uma das principais vantagens da utilização dos métodos *ab initio* em relação aos métodos por homologia (comparação das sequências com as de outras espécies já definidas) é que o primeiro permite a descoberta de novos genes [Zhu et al. 2010] que não possuem homólogos. Essa abordagem realiza uma busca sistemática nos dados de forma que seja possível realizar uma distinção entre uma região codificadora e uma região não codificadora. Uma das características mais usadas nessa abordagem é a

identificação de ORFs. Entretanto nem todo ORF é uma região codificadora [Angelova et al. 2010] e por isso utilizam-se outras características em conjunto, como a códon usage e o conteúdo de GC que já são usadas em diferentes ferramentas.

Este artigo apresenta o GeneFinder-TR, um *pipeline* para predição de genes em dados metatranscriptômicos baseado em um algoritmo *Random Forest*, que é um algoritmo robusto e por sua estrutura arbórea permitir encontrar dados significativos para classificação em conjuntos de dados menores [Maxion e Killourhy 2010]. O método provou ser eficiente quando aplicado a predição de genes em metagenomas [Góes et al. 2014] onde obteve os melhores resultados quando comparados aos classificadores. Além de usar o uso de códons e o conteúdo de GC como características para o processo de predição, ele também usa a frequência de aminoácidos presentes em cada sequência codificadora e não codificadora, uma vez que se considera que os genomas ricos em conteúdo de GC tendem a estar mais estreitamente ligados a Aminoácidos específicos [Bohlin et al. 2013]. O *pipeline* também usa valores para os 20 aminoácidos de acordo com cada uma das 544 propriedades físicas químicas do banco de dados AAindex, pois elas foram utilizadas com sucesso em estudos sobre métodos de predição e classificação [Blythe e Flower 2005], [Nanni e Lumini 2006]. Os resultados demonstram que o GeneFinder-TR tem um desempenho equiparável e até melhor que outras ferramentas para predição de gene quando tem como dados de entrada *reads* de diferentes tamanhos assim como de comunidades microbiais de diferentes complexidades. GeneFinder-TR é, dentre as ferramentas do experimento realizado, a única que utiliza os valores das propriedades físico-químicas como característica na busca de padrões para a classificação de sequências.

2. Pipeline e materiais

O aprendizado de máquina pode ser definido como um conjunto de métodos computacionais que utilizam experiências para realizar predições mais exatas. Em aprendizado de máquina supervisionado, métodos de classificação criam classificadores a fim de determinar o resultado de dados novos fornecidos a partir de um modelo criado por dados de treino usados anteriormente [Han et al. 2011].

GeneFinder-TR foi criado com base em um modelo de predição utilizando o pacote *caret* da linguagem de programação R. Ele foi baseado no classificador *Random Forest*, que são uma combinação de árvores de decisão em que cada uma das árvores é um modelo independente. Segundo [Han et al. 2011] uma árvore de decisão é um método de classificação que imita uma árvore na qual cada nó interno é um teste de um atributo, os galhos representam o resultado de um teste e cada nó final ou folha representa uma classe identificadora. Modelos baseados em árvore de decisão tendem a gerar *overfitting*, uma vez que apenas um modelo é gerado e o mesmo é bastante atrelado aos dados utilizados na fase de treino. O *Random Forest* utiliza a estratégia *ensemble*, que é uma estratégia que busca aliviar o problema de *overfitting*, através da combinação de vários modelos (diversas árvores de decisão). Nesse método, um conjunto de árvores de decisão cria uma “floresta” onde cada uma das árvores terá atributos randômicos e serão treinados com um conjunto de dados conhecidos, gerando um modelo que será utilizado para a classificação de dados de teste [Mohri et al. 2012].

O GeneFinder-TR utiliza as seguintes características das sequências para realizar a predição (1) o conteúdo de GC, (2) *códon usage*, (3) frequência dos aminoácidos

presentes em cada sequência codificadora e não codificadora – estudos ligam genomas ricos em conteúdo de GC estão mais ligados a aminoácidos específicos – e (4) os valores das 544 propriedades físico-químicas do banco AAindex. Estudos que utilizaram os valores destas propriedades para cada aminoácido presente obtiveram sucesso em métodos de classificação. As sequências tiveram os seus valores dados a partir de uma média de todos os valores das 544 propriedades para cada aminoácido da sequência. Por exemplo, a propriedade CHAM830102 possui os valores 0,124 para a Arginina, 0,289 para a Asparagina e 0,104 para a Glicina. Uma sequência que contém apenas uma ocorrência de cada um desses aminoácidos terá o valor aproximado de 0,172 para a propriedade CHAM830102.

O *Random Forest* criará um modelo para definir quais entradas são regiões codificadoras e quais são regiões não codificadoras baseando-se na extração das características das sequências de treinos fornecidas para criar um modelo de predição que possa ser aplicado aos dados de teste. Após a utilização dessas sequências de treino, o modelo gerado é aplicado às sequências de teste. Os conjuntos de treino e de teste foram divididos respectivamente em 70% e 30%. Foram escolhidas 250 árvores para a elaboração do modelo.

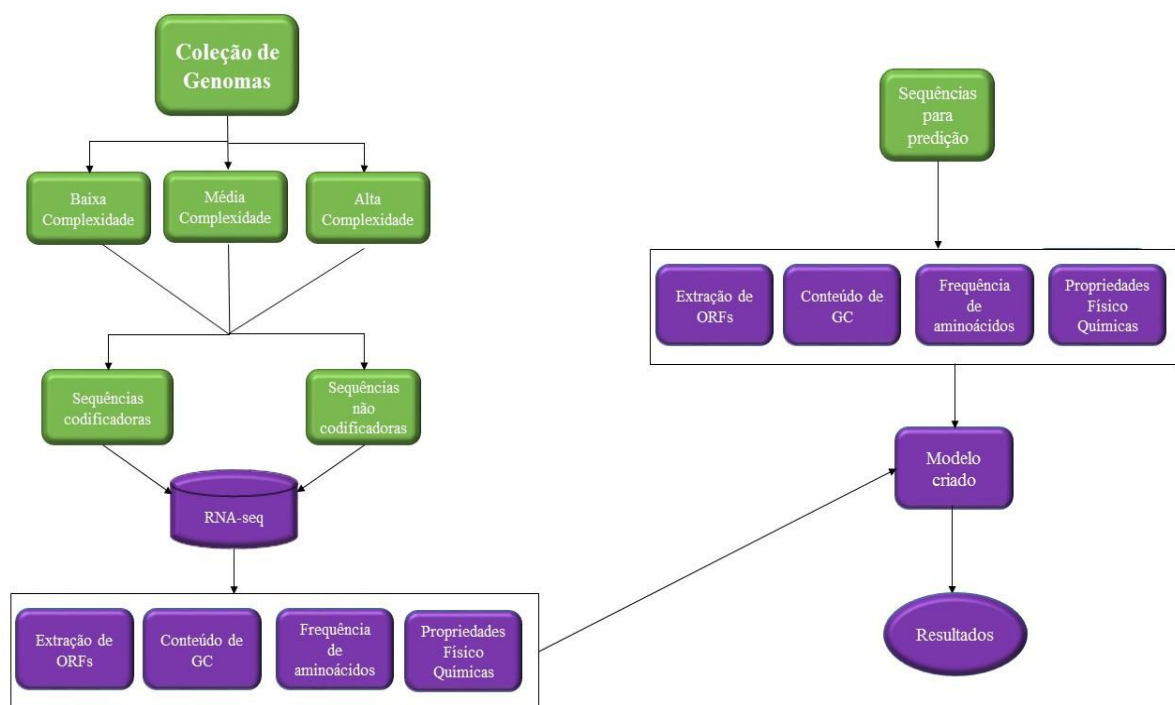


Figura 3. Diagrama do Pipeline proposto. Da esquerda para direita, os metagenomas são simulados em diferentes complexidades. As sequências passam pelo protocolo RNA-seq e têm as suas características extraídas e passadas para um classificador que criará um modelo que depois será utilizado para os dados de teste. Fonte: Autores.

2.1. Materiais

Os metagenomas usados para a comparação das ferramentas foram os mesmos presentes em [Mavromatis et al. 2007]. Esses metagenomas foram simulados a partir dos genomas de 113 espécies. Um metagenoma é dividido em níveis de complexidade. (1) Um

metagenoma que possui uma ou duas espécies dominantes é considerado de baixa complexidade; (2) um metagenoma que possui mais espécies dominantes é considerado de média complexidade; e (3) um metagenoma sem espécies dominantes é considerado de alta complexidade. Neste estudo o metagenoma de baixa complexidade teve uma única espécie dominante entre as amostras. O de média complexidade teve 5 espécies dominantes entre as amostras e o de alta complexidade não teve espécie dominante.

Polyester [Frazee et al. 2015] é uma ferramenta feita para simular o protocolo *RNA-seq*, que simula *reads* de transcritos que foram fornecidos como dados de entrada. Neste estudo, os dados simulados foram considerados a entrada para a ferramenta. Cada transcrito dos dados gera uma quantidade de *reads* determinada pelo usuário. O processo de sequenciamento do RNA é feito com a fragmentação – na qual os transcritos são divididos em pequenas unidades – e com o sequenciamento dos dados – onde os *reads* são gerados seguindo o protocolo Illumina *paired-end* [Sengupta et al. 2011]. Após o processo de sequenciamento, arquivos de extensão FASTA são criados contendo os *reads* criados e que serão lidos pelo algoritmo *Random Forest*. O conjunto de treino dos três metagenomas foi igualmente dividido em sequências codificadoras e não codificadoras para todos os organismos. Ambos os conjuntos tiveram sequências de tamanhos de 100bp, 200bp, 400bp e 800bp.

Orphelia, FragGeneScan, TransGeneScan e MetaGene foram usados em comparação com o GeneFinder-TR. Os *reads* gerados pelo Polyester foram comparados com as anotações dos genes para avaliar a acurácia de cada método de predição. Cada uma das ferramentas foi avaliada por sua acurácia, verificando a proporção de acerto na predição de sequências codificadoras e não codificadoras. Para este trabalho, as sequências são consideradas como codificadoras se ao menos 80% dela fizer o *overlap* (sobreposição) dos genes preditos e dos genes anotados no mesmo ORF. As sequências que tiverem menos que 80% são consideradas como falsos positivos.

3. Resultados e discussões

Foram analisados três metagenomas simulados com complexidades distintas. Os 113 genomas usados para simulação estão presentes nas bases do *Integrated Microbial Genomes* (IMG) e do *National Center for Biotechnology Information* (NCBI). Para comparação dos resultados foram observados os níveis de complexidade das amostras e o tamanho dos *reads*. Os resultados do GeneFinder-TR são comparados com os de TransGeneScan, FragGeneScan, Orphelia e MetaGene. Todas as ferramentas e o *pipeline* tiveram como entrada as mesmas sequências de *reads*.

A Tabela 1 e Figura 4 mostram os resultados para o metagenoma de baixa complexidade. O GeneFinder-TR teve os melhores resultados para *reads* de 100bp e de 800bp com acurácias de 92,53% e 95,14% respectivamente. TransGeneScan teve os melhores resultados para os *reads* de 200bp e 400bp com uma acurácia de 93,19% e 94,8% respectivamente.

Tabela 1. Resultados dos testes em metagenomas de baixa complexidade

simBC	Orphelia	MetaGene	FragGeneScan	TransGeneScan	GeneFinder-TR
100bp	0,9137	0,9031	0,9221	0,9218	0,9253

200bp	0,9287	0,9101	0,931	0,9319	0,9301
400bp	0,9303	0,9183	0,9388	0,948	0,9469
800bp	0,934	0,9198	0,9401	0,9511	0,9514

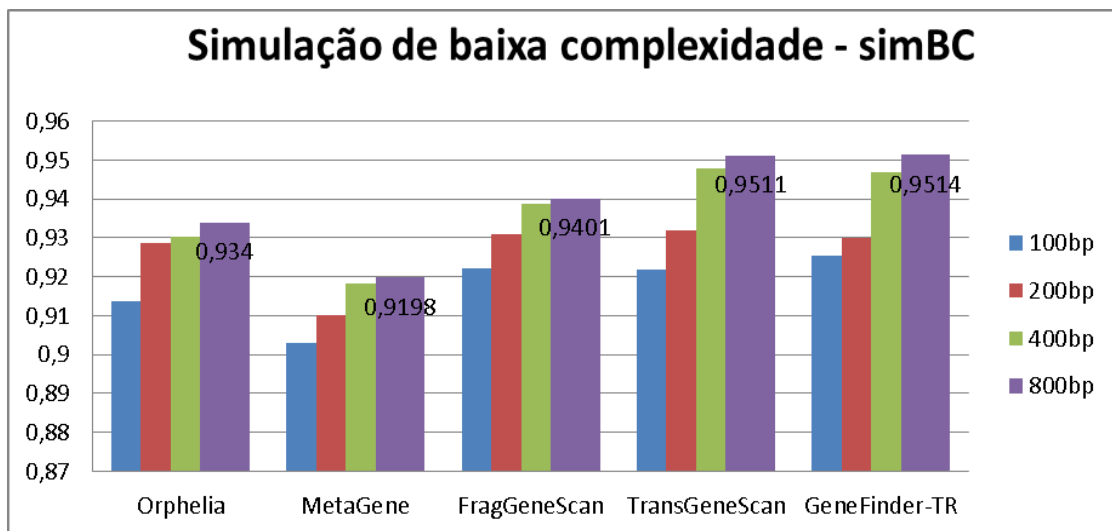


Figura 4. Resultados dos testes em metagenomas de baixa complexidade, mostrando a acurácia de cada umas das ferramentas de predição

A Tabela 2 e Figura 5 mostram os resultados para o metagenoma de média complexidade. Os resultados mostram que o FragGeneScan demonstrou a melhor acurácia para *reads* de 100bp com 92,33%. O GeneFinder-TR tem o melhor resultado para *reads* de 200bp com 93,01%. Já o TransGeneScan teve os melhores resultados para *reads* de 400bp com 93,97% e 800bp com 94,50%.

Tabela 2. Resultados dos testes em metagenomas de média complexidade

simMC	Orphelia	MetaGene	FragGeneScan	TransGeneScan	GeneFinder-TR
100bp	0,8991	0,8913	0,9232	0,9199	0,9213
200bp	0,9201	0,9019	0,9295	0,9278	0,9301
400bp	0,9289	0,9103	0,9347	0,9397	0,9387
800bp	0,9313	0,916	0,9381	0,945	0,9417

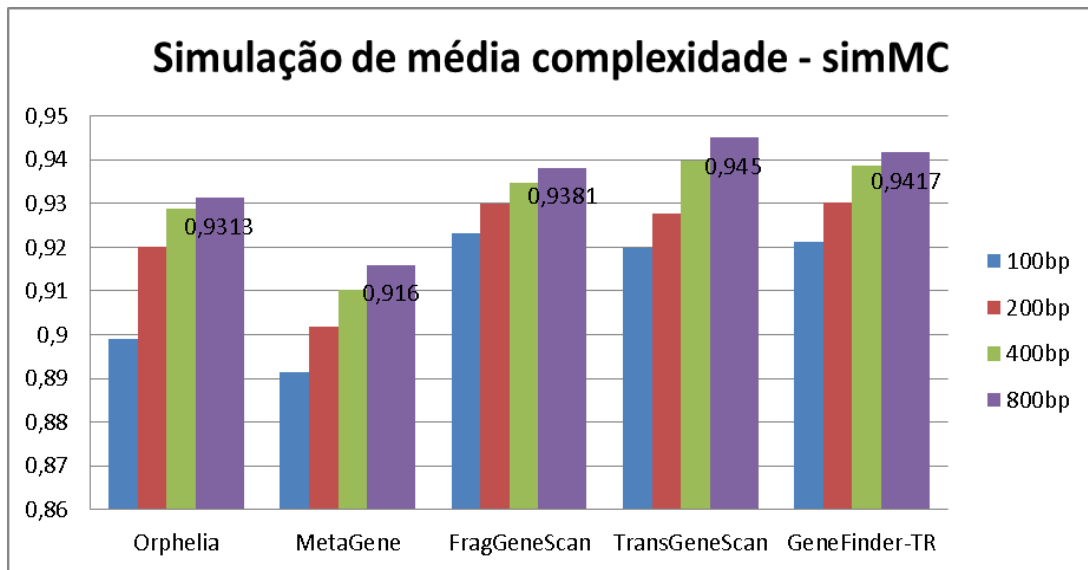


Figura 5. Resultados dos testes em metagenomas de média complexidade

A Tabela 3 e Figura 6 mostram os resultados para o metagenoma de alta complexidade. Os resultados mostram que o TransGeneScan tem o melhor resultado para *reads* de 200bp com 91,71%. O GeneFinder-TR teve os melhores resultados para os *reads* de 100bp com 91,16%, 400bp com 92,23% e 800bp com 93,91%.

Tabela 3. Resultados dos testes em metagenomas de alta complexidade

simAC	Orphelia	MetaGene	FragGeneScan	TransGeneScan	GeneFinder-TR
100bp	0,8923	0,8901	0,9031	0,9107	0,9116
200bp	0,913	0,8953	0,9099	0,9171	0,916
400bp	0,9171	0,9014	0,9131	0,9213	0,9223
800bp	0,9194	0,9101	0,9206	0,936	0,9391

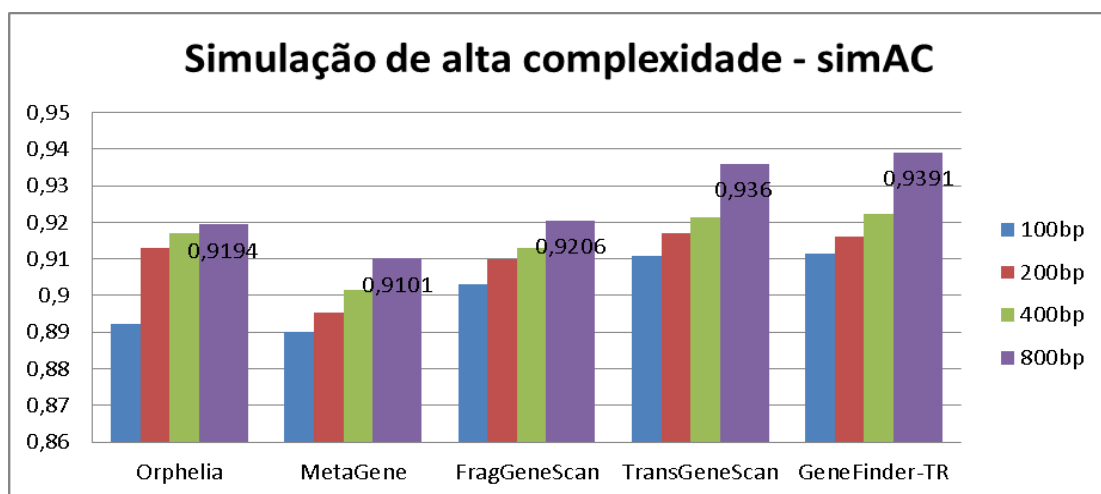


Figura 6. Resultados dos testes em metagenomas de alta complexidade

Os resultados mostraram que as ferramentas têm uma melhora de desempenho com o aumento do tamanho dos *reads*. Isso é explicado pelo fato de que a predição de genes é mais difícil de ser realizada em *reads* pequenos [Rho 2010]. As ferramentas Orphelia e MetaGene não obtiveram o melhor resultado nos testes realizados. O TransGeneScan demonstrou ser mais eficiente em metagenomas de baixa e média complexidade, enquanto o FragGeneScan teve seu melhor resultado em *reads* pequenos em um metagenoma de média complexidade. Essas ferramentas foram baseadas em um HMM com o intuito de melhorar a predição de genes em *reads* menores

GeneFinder-TR apresentou os melhores resultados nos metagenomas das três complexidades - baixa, média e alta - assim como em *reads* de 100bp, 200bp, 400bp e 800bp. Ele demonstrou ser eficiente na predição de genes em *reads* menores, um desafio para a predição de genes. Os resultados são animadores e demonstram que o *pipeline* se adapta a diferentes situações dos metagenomas estudados.

4. Conclusão

A predição de genes em dados de comunidades microbiais é um desafio para métodos computacionais. Muitas ferramentas foram desenvolvidas com o propósito de criar melhores métodos de predição de genes para esses dados. Poucas ferramentas, no entanto, trabalham com os dados de transcritos dessas comunidades. Neste artigo, apresentamos o GeneFinder-TR, um *pipeline* para predição de genes em dados metatranscriptômicos. Os resultados demonstraram que o *pipeline* teve os melhores resultados em metade dos testes realizados e resultados satisfatórios nos outros testes realizados para a predição de genes em comunidades de diferentes níveis de complexidade, e para *reads* de diferentes tamanhos. Foram realizadas comparações com outras ferramentas para predição de genes em dados metagenômicos e dados metatranscriptômicos em conjunto com o GeneFinder-TR para comparar o desempenho do *pipeline* em relação ao estado da Arte.

Os autores planejam estender o pipeline para a análise diferencial de expressão gênica. Até o presente momento os autores não encontraram ferramentas ou pipelines que façam, concomitantemente, a predição e análise diferencial de expressão gênica para dados metatranscriptômicos de diversas amostras utilizando o *Random Forest*.

Referências

- Angelova, M., Kalajdziski, S., and Kocarev, L. (2010). Computational methods for gene finding in prokaryotes. *ICT Innovations*, 11-20.
- Blythe, M. J., and Flower, D. R. (2005). Benchmarking B cell epitope prediction: underperformance of existing methods. *Protein Science*, 14(1), 246-248.
- Bohlin, J., Brynildsrud, O., Vesth, T., Skjerve, E., and Ussery, D. W. (2013). Amino acid usage is asymmetrically biased in AT-and GC-rich microbial genomes. *PLoS One*, 8(7), e69878.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
- Ercolini, D. (2013). High-throughput sequencing and metagenomics: moving forward in the culture-independent analysis of food microbial ecology. *Applied and environmental microbiology*, 79(10), 3148-3155.

- Frank, D. N., and Pace, N. R. (2008). Gastrointestinal microbiology enters the metagenomics era. *Current opinion in gastroenterology*, 24(1), 4-10.
- Frazeo, A. C., Jaffe, A. E., Langmead, B., and Leek, J. T. (2015). Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics*, 31(17), 2778-2784.
- Goés, F., Alves, R., Correa, L., Chaparro, C. and Thom, L., "Towards an ensemble learning strategy for metagenomic gene prediction." In the *Brazilian Symposium on Bioinformatics (BSB)*, Belo Horizonte, MG, Brazil, October 28-30, 2014. *Lecture notes in Bioinformatics (LNBI)*, vol. 8826 Springer, pp. 17-24.
- Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.
- Han, J., Kamber, M., and Pei, J. (2012). Classification: basic concepts. *Data mining Concepts and techniques*, Amsterdam: Elsevier.
- Hoff, K. J., Lingner, T., Meinicke, P., and Tech, M. (2009). Orphelia: predicting genes in metagenomic sequencing reads. *Nucleic acids research*, 37(suppl_2), W101-W105.
- Ismail, W. M., Ye, Y., and Tang, H. (2014). Gene finding in metatranscriptomic sequences. *BMC bioinformatics*, 15(9), S8.
- Kunin, V., Copeland, A., Lapidus, A., Mavromatis, K., and Hugenholtz, P. (2008). A bioinformatician's guide to metagenomics. *Microbiology and Molecular Biology Reviews* 72(4) 557-578
- Manichanh, C., Rigottier-Gois, L., Bonnaud, E., Gloux, K., Pelletier, E., Frangeul, L., ... and Roca, J. (2006). Reduced diversity of faecal microbiota in Crohn's disease revealed by a metagenomic approach. *Gut*, 55(2), 205-211.
- Mavromatis, K., Ivanova, N., Barry, K., Shapiro, H., Goltsman, E., McHardy, A. C., ... and Lapidus, A. (2007). Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. *Nature methods*, 4(6), 495-500.
- Maxion, Roy A., and Kevin S. Killourhy. "Keystroke biometrics with number-pad input." *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*. IEEE, 2010.
- Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012) *Foundations of Machine Learning*
- Nanni, L., and Lumini, A. (2006). An ensemble of K-local hyperplanes for predicting protein-protein interactions. *Bioinformatics*, 22(10), 1207-1210.
- Noguchi, H., Park, J., and Takagi, T. (2006). MetaGene: prokaryotic gene finding from environmental genome shotgun sequences. *Nucleic acids research*, 34(19), 5623-5630.
- Qin, J., Li, Y., Cai, Z., Li, S., Zhu, J., Zhang, F., ... and Peng, Y. (2012). A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, 490(7418), 55-60.
- Rho, M., Tang, H., and Ye, Y. (2010). FragGeneScan: predicting genes in short and error-prone reads. *Nucleic acids research*, 38(20), e191-e191.

- Rondon, M. R., August, P. R., Bettermann, A. D., Brady, S. F., Grossman, T. H., Liles, M. R., ... and Tjong, C. L. (2000). Cloning the soil metagenome: a strategy for accessing the genetic and functional diversity of uncultured microorganisms. *Applied and environmental microbiology*, 66(6), 2541-2547.
- Scheperjans, F., Aho, V., Pereira, P. A., Koskinen, K., Paulin, L., Pekkonen, E., ... and Kinnunen, E. (2015). Gut microbiota are related to Parkinson's disease and clinical phenotype. *Movement Disorders*, 30(3), 350-358.
- Seshadri, R., Kravitz, S. A., Smarr, L., Gilna, P., and Frazier, M. (2007). CAMERA: a community resource for metagenomics. *PLoS biology*, 5(3), e75.
- Stanke, M., Schöffmann, O., Morgenstern, B., and Waack, S. (2006). Gene prediction in eukaryotes with a generalized hidden Markov model that uses hints from external sources. *BMC bioinformatics*, 7(1), 62.
- Willner, D., Furlan, M., Haynes, M., Schmieder, R., Angly, F. E., Silva, J., ... and Rohwer, F. (2009). Metagenomic analysis of respiratory tract DNA viral communities in cystic fibrosis and non-cystic fibrosis individuals. *PloS one*, 4(10), e7370.
- Zhu, W., Lomsadze, A., and Borodovsky, M. (2010). Ab initio gene identification in metagenomic sequences. *Nucleic acids research*, 38(12), e132-e132.

Encoding the Consistency of Relational Bayesian Networks

Glauber De Bona^{1*}, Fabio G. Cozman¹

¹Escola Politécnica – Universidade de São Paulo (EP-USP)

Abstract. *Given an input structure, a relational Bayesian network returns a Bayesian network over ground atoms. In this paper, we analyze the problem of checking the consistency of a relational Bayesian network for a given class of input structures. This consistency is defined as the acyclicity of the output Bayesian network. We employ first-order logic to encode both the input structure and the structure of the corresponding output Bayesian network. Then we use a transitive closure operator to express the acyclicity of the resulting Bayesian network, translating the consistency of a relational Bayesian network for a given input class into the validity of a logical formula.*

1. Introduction

Computational systems targeting real-world problems usually need to reason both under uncertainty and about complex relational structures. On the one hand, probability theory is the standard mathematical tool to model uncertainty, and probabilistic graphical models, as Bayesian networks, efficiently encode probability distributions, allowing practicable inference and learning tasks. On the other hand, first-order logic is the formal framework to reason about objects and their relation, enabling one to represent general knowledge (or beliefs) over a domain. Consequently, several formalisms endowing probabilistic graphical models with first-order logic elements have been proposed in the Artificial Intelligence community in the past decades, under the field named as *statistical relational learning* [Getoor and Taskar 2007]. Instances of such approaches include Markov logic networks [Richardson and Domingos 2006], probabilistic-logic programming [Poole 1993, Sato 1995], probabilistic relational models [Koller 1999] and relational Bayesian networks [Jaeger 1997, Jaeger 2002]. The latter is the focus of the present work, as it can be regarded as at least as expressive as Markov logics [Jaeger 2008] and have probabilistic relational models as a particular case.

Relational Bayesian networks were proposed by Jaeger [Jaeger 1997, Jaeger 2002] in order to model probability distributions over predicates (or relations) whose domain is *a priori* unknown and not bounded. To grasp the expressive power of this framework, consider the following example.

Example 1 (*Scenario 1*) *Consider 3 neighbours, x_1, x_2, x_3 , who live close enough to hear each other's house alarm. As they live in a quite dangerous zone, each day there is a probability of 0.1% for each house to be robbed. When a house is robbed, there is probability of 90% that its alarm will go off. As sometimes happens, alarms can also ring without an apparent cause, so even without any burglary, an alarm rings at a given day with probability 1%. If an alarm goes off, all three neighbours immediately call the police to report the incident, and they do not call otherwise. Figure 1 depicts the Bayesian network modelling this situation, when we consider the dashed arcs.*

*The first author is supported by Fapesp (grant 2016/25928-4). The second author was partially supported by CNPq (grant 308433/2014-9) and the project has been funded by FAPESP (grant 2016/18841-0).

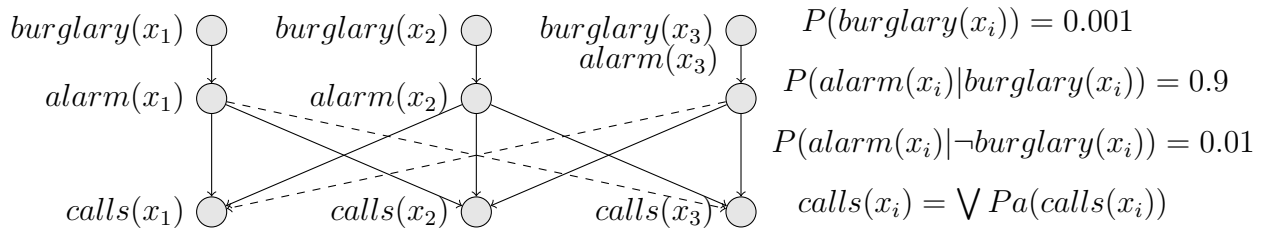


Figure 1. Bayesian network modelling the burglary-alarm-call situation with 3 people. $Pa(X)$ denotes the parents in the graph of the node X .

(Scenario 2) Now consider there are n neighbours x_1, x_2, \dots, x_n in the same situation. It is straightforward to expand the Bayesian network in Figure 1 to accommodate these n people. But if the number of neighbours is unknown, there is no fixed Bayesian networks that can allow for modelling the situation. Nevertheless, some kind of template for Bayesian networks can still capture this situation, as there is a fixed underlying pattern for the dependency arcs.

(Scenario 3) Suppose a slightly different situation with 3 people now: x_1 and x_2 are neighbours, so are x_2 and x_3 , but x_1 and x_3 are not neighbours. We assume that only neighbours can hear the alarm of each other. This means that x_1 calls the police if his own alarm or x_2 's goes off; the same happens with x_3 . But x_2 , the common neighbour, calls the police if any of the three alarms rings. This new situation is codified into the Bayesian network given in Figure 1 when we ignore the dashed arcs.

(Scenario 4) Imagine we want to generalize this new scenario for n people. Without knowing which pairs can be considered neighbours, there is no way we can predict in advance how the Bayesian network structure will be.

Relational Bayesian networks can capture exactly the type of situations like (Scenario 4) above, where the resulting Bayesian network depends not only on the domain size, but also on some relations within this domain. These relations are given in the form of a first-order structure as an input to a relational Bayesian network, whose output can be seen as a typical Bayesian network. Nevertheless, as the output Bayesian network depends on the relations that hold in the input structure, it may not be obvious in principle to check if it is a consistent – i.e., an acyclic – Bayesian network. More generally, one is interested in ensuring that input structures in a given class will always yield acyclic Bayesian networks, when the Bayesian network is defined as consistent for the given class. In this work, we encode the problem of checking consistency of a relational Bayesian network for a given class into a formula of a first-order logic with transitive closure operator.

After formally introducing relational Bayesian networks in Section 2, the consistency problem is formally defined in Section 3. In Section 4, we show how the output Bayesian network can be encoded into a first-order structure. We proceed to show how to encode the acyclicity of direct graphs by employing a transitive closure operator in Section 5. In Section 6, the consistency problem is translated into a logical formula.

2. Relational Bayesian Networks

In this section, we briefly introduce the formalism of relational Bayesian networks, in the version presented in [Jaeger 2002], which brings a thorough exposition on the topic.

Let S and R be disjoint sets of relation symbols, called the *predefined relations* and *probabilistic relations*, respectively. We assume that S contain the equality symbol $=$, to be interpreted in the usual way. Each predicate symbol is associated with a positive integer k , which is its *arity*. Given a finite domain $D = \{d_1, \dots, d_n\}$, if V is a set of relation symbols (as R or S), a V -structure \mathcal{D} is an interpretation of the symbols in V into sets of tuples in D . Formally, a V -structure \mathcal{D} maps each relation symbol $v \in V$ with arity k into a mathematical relation on D^k . We denote by $\text{Mod}_D(V)$ the set of all V -structures over a given finite domain D . Given a domain D , a $v \in V$ with arity k and a tuple $t \in D^k$, $v(t)$ is said to be a *ground V -atom*. A V -structure \mathcal{D} defines truth values for ground atoms: if v is mapped to a relation containing t , we say that $v(t)$ is *satisfied by \mathcal{D}* , which is denoted by $\mathcal{D} \models v(t)$. According to the syntax of first-order logic, we can construct formulas using a vocabulary of relations V , together with variables, quantifiers and Boolean connectives. We call these V -formulas, and their meaning is given by the first-order logic semantics, as usual, through the V -structures. We denote by $\varphi(x_1, \dots, x_m)$ a V -formula where x_1, \dots, x_k are free variables, in the usual sense. If φ is a V -formula and \mathcal{D} is a V -structure, $\mathcal{D} \models \varphi$ denotes that φ is satisfied by \mathcal{D} .

A *random relational structure model for S and R* is a partial function that takes an S -structure \mathcal{D} , over some finite domain D , and returns a probability distribution $P(\mathcal{D}) : \text{Mod}_D(R) \rightarrow [0, 1]$ over the R -structures. An example of random relational structure model would be the function in (Scenario 4) of Example 1 that receives an S -structure of neighbours and returns a joint probability distribution over ground atoms for *burglary*(\cdot), *alarm*(\cdot), *calls*(\cdot). Each single probability distribution $P(\mathcal{D})$ is called an *instance* of the random relational structure model.

Relational Bayesian networks provide a way to compactly represent random relational structure models. This is achieved by mapping each S -structure into a *ground Bayesian network* that encodes a probability distribution over R -structures. This ground Bayesian network has nodes representing $r(t)$ (ground atoms), for each $r \in R$ and $t \in D^k$, where k is the arity of r . Thus, given the domain D of the input S -structure, the nodes in the corresponding Bayesian network are already determined. To define the arcs and parameters of the Bayesian network associated with an arbitrary S -structure, relational Bayesian networks employ their central notion of probability formula.

Probability formulas are syntactical constructions intended to link the probability of a ground atom $r(t)$ to the probabilities of other ground atoms $r'(t')$, according to the S -structure. Fixed an R -structure and an S -structure, for elements t_1, \dots, t_k in the domain D , a probability formula $F(t_1, \dots, t_k)$ should evaluate to a number in $[0, 1]$. The definition of probability formulas makes use of *combination functions*, which are functions from finite multi-sets over the interval $[0, 1]$ to numbers in the same interval. We use $\{\cdot\}$ to denote multi-sets. For instance, noisy – or is a combination function such that, if $c_1, \dots, c_n \in [0, 1]$, noisy – or $\{\{c_1, \dots, c_n\}\} = \prod_{i=1}^n 1 - (1 - c_i)$.

Definition 1 Given disjoint sets S and R of relation symbols and a tuple x of k variables, $F(x)$ is a (S, R) -probability formula if

- (constants) $F(x) = c$ for a $c \in [0, 1]$;
- (indicator functions) $F(x) = r(x)$, for an $r \in R$ with arity k ;
- (convex combinations) $F(x) = F_1(x)F_2(x) + (1 - F_1(x))F_3(x)$, where

- $F_1(x), F_2(x), F_3(x)$ are probability formulas, or;
- (combination functions) $F(x) = \text{comb}\{F_1(x, y), \dots, F_m(x, y)|y; \varphi(x, y)\}$, where comb is a combination function, $F_1(x, y), \dots, F_m(x, y)$ are probability formulas, y is a tuple of variables and $\varphi(x, y)$ is an S -formula.

Relational Bayesian networks associate a probability formula $F_{r^*}(x)$ to each probabilistic relation $r^* \in R$, where x is a tuple of k variables, the arity of r^* :

Definition 2 Given disjoint sets of relation symbols S and R , the predefined and probabilistic relations, a *Relational Bayesian Network* is a set $\Phi = \{F_r(x) \mid r \in R\}$, where each $F_r(x)$ is a (S, R) -probability formula.

To have an idea of how probability formulas work, consider a fixed S -structure \mathcal{D}_S over a domain D . Then, an R -structure \mathcal{D}_R over D entails a numeric value for each *ground probability formula* $F_{r^*}(t)$, denoted by $F_{r^*}(t)[\mathcal{D}_R]$, where t is tuple of elements in D . This is done inductively, by initially defining $r(t)[\mathcal{D}_R] = 1$ if $\mathcal{D}_R \models r(t)$, and $r(t)[\mathcal{D}_R] = 0$ otherwise, for each $r(t)$, for all $r \in R$. If $F_{r^*}(x) = c$, then $F_{r^*}(t)[\mathcal{D}_R] = c$, for any tuple t . The numeric value of $F_{r^*}(t)\mathcal{D}_R$ for probability formulas that are convex combinations or combination function will require the evaluation of its subformulas F_i , which recursively end at the evaluation of ground atoms $r(t)$ or constants c . As the set of ground atoms whose evaluation is needed to compute $F_{r^*}(t)[\mathcal{D}_R]$ depends only on the S -structure \mathcal{D}_S , and not on \mathcal{D}_R , it is denoted by $\alpha(F_{r^*}(x), t, \mathcal{D}_S)$ and can be defined recursively:

- $\alpha(c, t, \mathcal{D}_S) = \emptyset$;
- $\alpha(r(x), t, \mathcal{D}_S) = \{r(t)\}$;
- $\alpha(F_1(x)F_2(x) + (1 - F_1(x))F_3(x), t, \mathcal{D}_S) = \bigcup_{i=1}^3 \alpha(F_i(x), t, \mathcal{D}_S)$;
- $\alpha(\text{comb}\{F_1(x, y), \dots, F_m(x, y)|y; \varphi(x, y)\}, t, \mathcal{D}_S)$ is given by
$$\bigcup_{t' \text{ s.t. } \mathcal{D}_S \models \varphi(t, t')} \bigcup_{i=1}^m \alpha(F_i(x, y), (t, t'), \mathcal{D}_S).$$

Above (t, t') denotes the concatenation of the tuples t and t' .

For a given S -structure \mathcal{D}_S , we can define a dependency relation between the nodes $r(t)$ and $r'(t')$ in the Bayesian network via the corresponding probability formulas F_r and $F_{r'}$ by employing the corresponding $\alpha(\cdot, \cdot, \cdot)$. Intuitively, $\alpha(F_r(x), t, \mathcal{D}_S)$ contains the ground atoms $r'(t')$ whose truth value in a structure \mathcal{D}_R determines the value of $F_r(t)$, which is meant to be the probability of $r(t)$. That is, $\alpha(F_r(x), t, \mathcal{D}_S)$ contains the parents of $r(t)$. Formally, we define the relation \preceq , over ground R -atoms, such that $r(t) \preceq r'(t')$ iff $r'(t') \in \alpha(F_r(x), t, \mathcal{D}_S)$. When this relation is acyclic, a relational Bayesian network $\Phi = \{F_r \mid r \in R\}$ defines, for a given S -structure \mathcal{D}_S over a finite domain D , a probability distribution over the R -structures \mathcal{D}_R over D via:

$$P_{\mathcal{D}_S}^{\Phi}(\mathcal{D}_R) = \prod_{r \in R} \prod_{t, \mathcal{D}_R \models r(t)} F_r(t)[\mathcal{D}_R] \prod_{t, \mathcal{D}_R \not\models r(t)} (1 - F_r(t)[\mathcal{D}_R])$$

Example 2 Recall the situation in the (Scenario 4) of Example 1. We can define a relational Bayesian network that returns a Bayesian network for each number and configuration of neighbours. Let $S = \{\text{neighbour}(\cdot, \cdot)\}$ and $R =$

¹Note that $F_{r^*}(x)$ cannot mention another $F_{r'}(x')$, but has to be inductively built from constants and indicator functions with convex combinations and combination functions; otherwise the dependency relation \preceq (to be defined) would be cyclic, yielding inconsistency.

$\{burglary(\cdot), alarm(\cdot), calls(\cdot)\}$. We assume that the relation *neighbour* is reflexive symmetrical. For each relation in R , we associate a probability formula, forming the relational Bayesian network Φ :

- $F_{burglary}(x) = 0.001$; a constant;
- $F_{alarm}(x) = burglary(x)0.9 + 0.1(1 - burglary(x))0.1$; (nested) convex combinations;
- $F_{call}(x) = noisy - or\{alarm(y) \mid y; neighbour(x, y)\}$; a combination function.

Note that $1 - F_1$ and F_1F_2 are convex combinations and, thus, probability formulas. As the inputs of the noisy – or above are in $\{0, 1\}$, it actually works like a Boolean or.

Given an S -structure \mathcal{D}_S over a domain D , Φ determines a joint probability distribution over the ground R -atoms, via a Bayesian network. If we take an S -structure \mathcal{D}_S over a domain $D = \{x_1, x_2, x_3\}$ such that $\mathcal{D}_S \models neighbour(x_1, x_2) \wedge neighbour(x_2, x_3)$ but $\mathcal{D}_S \not\models neighbour(x_1, x_3)$, the resulting $P_{\mathcal{D}_S}^\Phi$ is the model of (Scenario 3) in Example 1, whose Bayesian network is given in Figure 1 when we ignore the dashed arcs.

3. Defining the Consistency Problem

For a relational Bayesian network Φ , not all S -structures yield an acyclic dependency relation \preceq . When the relation \preceq is cyclic, the resulting Bayesian network is cyclic, and no probability distribution is defined over the R -structures. In such cases, we say Φ is *inconsistent* for that S -structure. This notion can be generalized to a class of S -structures \mathcal{S} , so that we say that Φ is consistent for \mathcal{S} iff the resulting relation \preceq is acyclic for each S -structure in \mathcal{S} . To know if a relational Bayesian network is consistent for given class of S -structures is precisely the problem we are interested in this work.

In order to reason about the relation between a class of S -structures and the consistency of a relational Bayesian network Φ for it, we need to formally represent these concepts. To define a class of S -structures, we can remind that they can be seen as first-order structures over which S -formulas are interpreted. That is, an S -formula defines the set of S -structures satisfying it. If φ is a closed S -formula (without free variables), we say that $[[\varphi]]_S$ is the set of S -structures \mathcal{D}_S such that $\mathcal{D}_S \models \varphi$. We denote by θ_S an S -formula satisfied exactly by the S -structures in a given class \mathcal{S} ; that is such that $[[\theta_S]] = \mathcal{S}$.

To encode the consistency of Φ , we need to encode the acyclicity of the dependency relation \preceq resulting from an S -structure. Ideally, we would like to have a (first-order) S -formula, say ψ_Φ , that is true only for S -structures yielding acyclic dependency relations \preceq . If that were the case, to decide the consistency of Φ for the class \mathcal{S} would be reduced to decide the validity of the first-order formula $\theta_S \rightarrow \psi_\Phi$. If this formula is valid, then every S -structure in the class \mathcal{S} guarantees that the resulting dependency relation \preceq for Φ is acyclic, hence Φ is consistent for \mathcal{S} . Otherwise, there is an S -structure in \mathcal{S} yielding a cyclic dependency relation \preceq for Φ ². Unfortunately, to encode cycles in a graph, one needs to encode the notion of path, which is the transitive closure of a relation encoding arcs, but it is a well-known fact that first-order logic cannot express transitivity. To circumvent that, we can add a transitive closure operator to the logic, arriving at the so called transitive closure logics, as described for instance in [Alechina and Immerman 2000].

²Note that for S -formulas only S -structures matters, and we could ignore any relation not in S . To be precise, if a first-order structure \mathcal{D} falsifies $\theta_S \rightarrow \psi_\Phi$, then there is an S -structure \mathcal{D}_S (formed by ignoring non- S relations) falsifying it.

This approach was put forward by Jaeger [Jaeger 2002], who assumed one could write down the S -formula ψ_Φ by employing a transitive closure operator. He conjectured that with some restrictions on the arity of the relations in S and R one could hope to obtain a formula $\theta_S \rightarrow \psi_\Phi$ that is decidable. Nevertheless, no hint was provided as to how to construct such a formula, or as to its general aspect. A main difficulty is that, if an S -structure \mathcal{D} satisfying θ_S has domain $D = \{d_1, \dots, d_n\}$, the size of the resulting Bayesian network is typically greater than n , with one node per ground atom, so a cycle can also contain more nodes than n . Thus, there is no direct way of employing the transitive closure operator to devise a formula $\neg\psi_\Phi$, encoding cycles with more than n nodes, that is to be satisfied by some structures \mathcal{D} over a domain with only n elements. In the next section, we explore a path to encode ψ_Φ for an augmented domain, through an auxiliary formula whose satisfying structures will represent both the S -structure and the resulting ground Bayesian network. Afterwards, we adapt the formula θ_S accordingly.

4. Encoding the Structure of the Ground Bayesian Network

Our idea to construct a formula ψ_Φ , for a given relational Bayesian network Φ , is first to find a first-order V -formula \mathcal{B}_Φ , for some vocabulary V containing S , that is satisfiable only by V -structures that encode both an S -structure \mathcal{D}_S and the structure of the ground Bayesian network resulting from it. These V -structures should contain, besides an S -structure \mathcal{D}_S , an element for each node in the ground Bayesian network and a relation capturing its arcs. Then we can use a transitive closure operator to define the existence of paths (and cycles) via arcs, for enforcing acyclicity by negating the existence of a cycle.

Suppose we have two disjoint vocabularies S and $R = \{r_1, \dots, r_m\}$ of predefined and probabilistic relations, respectively. We use $a(v)$ to denote the arity of a relation v . Consider a relational Bayesian network $\Phi = \{F_r(x) \mid r \in R\}$, where each $F_r(x)$ is a (S, R) -probability formula. Let \mathcal{D} be a V -structure satisfying \mathcal{B}_Φ . We want \mathcal{D} to be defined over a bipartite domain $D = D_S \cup D_B$, where D_S is used to represent an S -structure \mathcal{D}_S and $D_B = D \setminus D_S$ is the domain where the structure of the resulting ground Bayesian network is encoded. We include in V a unary predicate $D_S(\cdot)$ that shall be true for all and only the elements in D_S . The structure \mathcal{D} shall represent the structure of the ground Bayesian network $B_\Phi(\mathcal{D}_S)$, over the elements of D_B , that is induced by the S -structure \mathcal{D}_S codified in D_S . In order to accomplish that, \mathcal{D} must have an element in D_B for each ground atom over the domain D_S . Furthermore, the V -structure \mathcal{D} must interpret a relation, say $Parent(\cdot, \cdot)$, over D_B according to the arcs of the Bayesian network $B_\Phi(\mathcal{D}_S)$.

Firstly, we need to define a vocabulary V , which includes the predefined relations S^3 and contains the unary predicate D_S . Furthermore, V must contain a binary relation $Parent$ to represent the arcs of the ground Bayesian network. As auxiliary relations for defining $Parent$, we will need a relation Dep_i^j , for each pair $r_i, r_j \in R$, whose arity is $a(r_i) + a(r_j)$. For elements in D_B to represent ground atoms $r(t_1, \dots, t_n)$, we use relations to associate elements in D_B to relations r and to tuples $\langle t_1, \dots, t_n \rangle$. For each relation $r_i \in R$, we have a unary relation $\bar{r}_i \in V$, where $\bar{r}_i(x)$ is intended to mean that the element $x \in D_B$ represents a ground atom of the form $r_i(\cdot)$. As for the tuples, recall that each t_i represents an element in the set D_S over which the S -structure \mathcal{D}_S is codified. Hence, we insert in V binary relations t_i for every $1 \leq i \leq \max_i a(r_i)$, such that $t_i(x, y)$

³Recall that the equality symbol ($=$) is included in S .

should be true iff the element $x \in D_B$ corresponds to a ground atom $r(t_1, \dots, t_k)$ where $t_i = y$, for a $y \in D_S$ and some $r \in R$. To save notation, we use $R_i(x, y_1, \dots, y_k)$ to denote $\bar{r}_i(x) \wedge t_1(x, y_1) \wedge \dots \wedge t_k(x, y_k)$ henceforth, meaning the element x in the domain represents the ground atom $r_i(y_1, \dots, y_k)$, where $a(r_i) = k$.

Now we proceed to list, step-by-step, the set of conjuncts required in ψ_Φ , together with their meaning, for the V -structure \mathcal{D} in $[[\psi_\Phi]]$ to hold the desired properties.

4.1. Encoding the Nodes of the Bayesian network

We have to ensure that the elements in D_B correspond exactly to the ground atoms in the ground Bayesian network $B_\Phi(\mathcal{D}_S)$.

- Each element in $D_B = D \setminus D_S$ should correspond to a ground atom for some $r_i \in R$. Hence, we have the formula:

$$\forall x \neg D_S(x) \rightarrow \bigvee_{i=1}^m \bar{r}_i(x) \quad (1)$$

- No element may correspond to ground atoms for two different $r_i \in R$. Therefore, the formula below is employed:

$$\forall x \bigwedge_{1 \leq i, j \leq m}^{i \neq j} (\neg \bar{r}_i(x) \vee \neg \bar{r}_j(x)) \quad (2)$$

- Each element corresponding to a ground atom should correspond to exactly one tuple. To achieve that, let $k = \max_j a(r_j)$ and consider the formula below:

$$\forall x \forall y \forall z \bigwedge_{j=1}^k (t_j(x, y) \wedge t_j(x, z) \rightarrow y = z) \quad (3)$$

- Each element corresponding to a ground atom for a $r_i \in R$ should be linked a tuple with arity $a(r_i)$. Thus, let $k = \max_j a(r_j)$ and consider the formula below for each $r_i \in R$:

$$\forall x \bar{r}_i(x) \rightarrow (\exists y_1 \dots \exists y_{a(r_i)} R_i(x, y_1, \dots, y_{a(r_i)}) \wedge \forall z \neg t_{a(r_i)+1}(x, z) \wedge \dots \wedge \neg t_k(x, z)) \quad (4)$$

- Only elements in $D_B = D \setminus D_S$ should correspond to ground atoms. This is enforced by the following formula, where $k = \max_i a(r_i)$:

$$\forall y D_S(y) \rightarrow (\bigwedge_{i=1}^m \neg \bar{r}_i(y) \wedge \forall x \bigwedge_{j=1}^k \neg t_j(y, x)) \quad (5)$$

- Each ground atom must be represented by at least one element (in $D_B = D \setminus D_S$). So, for each $r_i \in R$, with $a(r_i) = k$, we need a formula:

$$\forall y_1 \dots \forall y_k D_S(y_1) \wedge \dots \wedge D_S(y_k) \rightarrow \exists x R_i(x, y_1, \dots, y_k) \quad (6)$$

These formulas enforce that each ground atom $r(t)$ is represented by an element x that is in D_B , due to the formula (5).

- No ground atom can be represented by two different elements. Hence, for each $r_i \in R$, with $a(r_i) = k$, we employ a formula:

$$\forall y_1, \dots, \forall y_k \forall x \forall z R_i(x, y_1, \dots, y_k) \wedge R_i(z, y_1, \dots, y_k) \rightarrow x = z \quad (7)$$

The conjunction of all formulas in (1)–(7) is satisfied only by structures \mathcal{D} over the domain $D = D_S \cup D_B$ such that there is a bijection between D_B and the set of all possible ground atoms $\{r(t) \mid \text{for some } r \in R \text{ and } t \in D_S^{a(r)}\}$. Now we can put the arcs over these nodes to complete the structure of the ground Bayesian network $B_\Phi(\mathcal{D}_S)$.

4.2. Encoding the Arcs of the Bayesian network

The binary relation *Parent* must hold only between elements in the domain D representing ground atoms $r(t)$ and $r'(t')$ such that $r(t) \preceq r'(t')$. Recall that the dependency relation \preceq is determined by the S -structure \mathcal{D}_S . While the ground atoms represented in D_B , for a fixed R , are determined by the size of D_S by itself, the relation *Parent* between them depends also on the S -formulas that hold for the S -structure \mathcal{D}_S . We want these S -structures to be specified by \mathcal{D} over D_S only, not over D_B . To ensure this, we use the following group of formulas:

- For all $s \in S$, consider the formula below, where $a(s) = k$:

$$\forall y_1 \dots \forall y_k s(y_1, \dots, y_k) \rightarrow D_S(y_1) \wedge \dots \wedge D_S(y_k) \quad (8)$$

The formula above forces that $s(t)$, for any $s \in S$, can be true only for tuples $t \in D_S^{a(s)}$.

For a known S -structure \mathcal{D}_S , it is straightforward to determine which ground atoms $r'(t')$ are the parents of $r(t)$ in the ground Bayesian network $B_\Phi(\mathcal{D}_S)$. One can simply use recursively the definition of the set of parents $\alpha(F_r(x), t, \mathcal{D}_S)$ given in Section 2. Nonetheless, with an unknown S -structure \mathcal{D}_S specified in \mathcal{D} over D_S , the situation is a bit trickier. The idea is to construct, for each pair $r_i(t)$ and $r_j(t')$, an S -formula $Dep_i^j(t, t')$ that is true iff $r_i(t) \preceq r_j(t')$ for the \mathcal{D}_S encoded in \mathcal{D} . To define $Dep_i^j(t, t')$, we employ auxiliary formulas $C_{F(t)}^{r'(t')}$, for a ground probability formula $F(t)$ and a ground atom $r'(t')$, that will be an S -formula that is satisfied by \mathcal{D} iff $r'(t') \in \alpha(F(x), t, \mathcal{S})$. We define $C_{F(t)}^{r'(t')}$ recursively, starting from the base cases.

- If $F(t) = c$, for a $c \in [0, 1]$, then $C_{F(t)}^{r'(t')} = \perp$.
- If $F(t) = r''(t)$, then $C_{F(t)}^{r'(t')} = (t' = t)$ if $r' = r''$; and $C_{F(t)}^{r'(t')} = \perp$ otherwise.

Above, $(t' = t)$ is a short form for $(t'_1 = t_1) \wedge \dots \wedge (t'_k = t_k)$, where k is the arity of t . These base cases are in line with the recursive definition of $\alpha(F(x), t, \mathcal{S})$ presented in Section 2. The third case is also straightforward:

- If $F(t) = F_1(t)F_2(t) + (1 - F_1(t))F_3(t)$, then $C_{F(t)}^{r'(t')} = \bigvee_{i=1}^3 C_{F_i(t)}^{r'(t')}$.

In other words, the computation of $F(t)[\mathcal{D}_R]$ depends on $r'(t')[\mathcal{D}_R]$, for some \mathcal{D}_R , if the computation of some $F_i(t)[\mathcal{D}_R]$, for $1 \leq i \leq 3$, depends on $r'(t')[\mathcal{D}_R]$.

The more elaborated case happens when $F(x)$ is a combination function, for there is an S -formula involved. Recall that if $F(x) =$

$\text{comb}\{F_1(x, y), \dots, F_m(x, y) | y; \varphi(x, y)\}$, then the parents of $F(t)$ are given by $\bigcup_{t', \mathcal{D}_S \models \varphi(t, t')} \bigcup_{i=1}^m \alpha(F_i(x, y), (t, t'), \mathcal{D}_S)$. Thus, to recursively define $C_{F(t)}^{r'(t')}$, we need an S -formula that is satisfied by an S -structure \mathcal{D}_S iff $r'(t') \in \bigcup_{t^*, \mathcal{D}_S \models \varphi(t, t^*)} \bigcup_{i=1}^m \alpha(F_i(x, y), (t, t^*), \mathcal{D}_S)$. The inner union is analogous to the definition of $C_{F(t)}^{r'(t')}$ for convex combinations. But to cope with any t^* such that $\mathcal{D}_S \models \varphi(t, t^*)$, we need an existential quantification:

- If $F(x) = \text{comb}\{F_1(x, y), \dots, F_m(x, y) | y; \varphi(x, y)\}$, then we have that $C_{F(t)}^{r'(t')} = \exists t^* \varphi(t, t^*) \wedge \bigvee_{i=1}^m C_{F_i(t, t^*)}^{r'(t')}$.

Now we can employ the formulas $C_{F(t)}^{r'(t')}$ to define the truth value of the ground relation $Dep_i^j(t, t')$, that codifies when $r_i(t) \preceq r_j(t')$.

- For each pair $r_i, r_j \in R$, with $a(r_i) = k$ and $a(r_j) = k'$, we have the formula:

$$\forall x_1 \dots \forall x_k \forall y_1 \dots \forall y_{k'} Dep_i^j(x_1, \dots, x_k, y_1, \dots, y_{k'}) \leftrightarrow C_{F_{r_i}(x_1, \dots, x_k)}^{r_j(y_1, \dots, y_{k'})} \quad (9)$$

In the formula above, $C_{F_{r_i}(x_1, \dots, x_k)}^{r_j(y_1, \dots, y_{k'})}$ has free variables $x_1, \dots, x_k, y_1, \dots, y_{k'}$ and is constructed according to the four recursive rules that define $C_{F(t)}^{r'(t')}$, replacing the tuples t and t' by x and y . We point out that such construction depends only on probability formulas in the relational Bayesian network Φ , and not on any S -structure. To build each $C_{F_{r_i}(x)}^{r_j(y)}$, one just starts from the probability formula $F_{r_i}(x)$ and follows the recursion rules until reaching the base cases, when $C_{F_{r_i}(x)}^{r_j(y)}$ will be formed by subformulas like \top, \perp , S -formulas $\varphi(\cdot)$ and equalities $(\cdot = \cdot)$, possibly quantified on variables appearing in φ .

The relation $Parent(\cdot, \cdot)$ is defined now over elements that represent ground atoms $r_i(t)$ and $r_j(t')$ such that $Dep_i^j(t, t')$, meaning that $r_i(t) \preceq r_j(t')$.

- For each pair $r_i, r_j \in R$, with $a(r_i) = k$ and $a(r_j) = k'$, let y and y' denote y_1, \dots, y_k and $y'_1, \dots, y'_{k'}$, respectively:

$$\forall x \forall x' \forall y_1 \dots \forall y_k \forall y'_1 \dots \forall y'_{k'} R_i(x, y) \wedge R_j(x', y') \wedge Dep_i^j(y, y') \leftrightarrow Parent(x, x') \quad (10)$$

Definition 3 Given disjoint sets of relations S and R and a relational Bayesian network $\Phi = \{F_{r_i} \mid r_i \in R\}$, the formula \mathcal{B}_Φ can be defined as the conjunction of all formulas in (1)–(10).

For some fixed relational Bayesian networks Φ , the formula \mathcal{B}_Φ is satisfied only by V -structures \mathcal{D} over a bipartite domain $D_S \cup D_B$ such that:

- the relations in S are interpreted in D_S , forming an S -structure \mathcal{D}_S ;
- there is a bijection b between the the domain $D_B = D \setminus D_S$ set of all ground R -atoms formed by the tuples in D_S ;
- each $x \in D_B$ is linked exactly to one $r_i \in R$, via the predicate $\bar{r}_i(x)$, and exactly $k = a(r_i)$ elements in D_S , via the relations $t_1(x, \cdot), \dots, t_k(x, \cdot)$, and no ground atom is represented through these links twice ;
- the relation $Parent(\cdot, \cdot)$ is interpreted as arcs in D_B in such a way that $\langle D_B, Parent \rangle$ form a directed graph that is the structure of the ground Bayesian network $B_\Phi(\mathcal{D}_S)$.

5. Encoding Acyclicity

The original formula ψ_Φ was intended to capture the consistency of the relational Bayesian network Φ . Our idea is to check the consistency by looking for cycles in the ground Bayesian network $B_\Phi(\mathcal{D}_S)$ encoded in any V -structure satisfying \mathcal{B}_Φ . Hence, we replace ψ_Φ by an implication $\mathcal{B}_\Phi \rightarrow \psi'_\Phi$, which is to be satisfied only by V -structures \mathcal{D} such that, if \mathcal{D} represents an S -structure \mathcal{D}_S and the resulting ground Bayesian network $B_\Phi(\mathcal{D}_S)$, then $B_\Phi(\mathcal{D}_S)$ is acyclic. Thus, ψ'_Φ should avoid cycles of the relation *Parent* in the V -structures satisfying it.

There is a cycle with *Parent*-arcs in a V -structure \mathcal{D} over a domain D iff there exists a $x \in D$ such that there is a path of *Parent*-arcs from x to itself. Consequently, detecting *Parent*-cycles reduces to compute *Parent*-paths, or *Parent*-reachability. We say y is *Parent*-reachable from x , in a V -structure \mathcal{D} , if there are $z_0, \dots, z_k \in D$ such that $x = z_0$, $y = z_k$, and $\mathcal{D} \models \bigwedge_{1 \leq i \leq k} \text{Parent}(z_{i-1}, z_i)$. Thus, for each k , we can define reachability through k *Parent*-arcs: $\text{Parent-Path}_k(x, y) = \exists z_0 \dots \exists z_k (z_0 = x) \wedge (z_k = y) \wedge \bigwedge_{1 \leq i \leq k} \text{Parent}(z_{i-1}, z_i)$. Unfortunately, the size of the path (k) is unbounded *a priori*, as the domain D can be arbitrarily large. Therefore, there is no means in the first-order logic language to encode reachability, via arbitrarily large paths, with a finite number of formulas. In order to circumvent this situation, we can resort to a transitive closure logic.

Transitive closure logics enhance first-order logics with a transitive closure operator TC – see e.g. [Alechina and Immerman 2000]. If $\varphi(x, y)$ is a first-order formula, $\text{TC}(\varphi)(x, y)$ means that y is φ -reachable from x . Accordingly, a V -structure \mathcal{D} , over a domain D , satisfies $\text{TC}(\varphi)(x, y)$ iff there is a $k \in \mathbb{N}$ and there are $z_0, \dots, z_k \in D$ such that $x = z_0$, $y = z_k$, and $\mathcal{D} \models \bigwedge_{1 \leq i \leq k} \varphi(z_{i-1}, z_i)$. Employing the transitive closure operator, the existence of a *Parent*-path from a node x to itself (a cycle) can be encoded directly by $\text{TC}(\text{Parent})(x, x)$; and the absence of a *Parent*-cycle can be enforced by $\psi'_\Phi = \forall x \neg \text{TC}(\text{Parent})(x, x)$.

The V -structures \mathcal{D} over a domain D satisfying $\mathcal{B}_\Phi \rightarrow \psi'_\Phi$ have the following format:

- either it is not the case that \mathcal{D} encodes both an S -structure in $D_S \subseteq D$ (the part of the domain satisfying $D_S(\cdot)$) and the corresponding ground Bayesian network $B_\Phi(\mathcal{D}_S)$ in $D_B = D \setminus D_S$;
- or it does encode an S -structure in $D_S \subseteq D$ and the corresponding *acyclic* ground Bayesian network $B_\Phi(\mathcal{D}_S)$ in $D_B = D \setminus D_S$.

6. Encoding the Consistency-Checking Problem

Back to the consistency-checking problem, we need to decide, for a fixed relational Bayesian network Φ , whether or not a given class \mathcal{S} of S -structures ensures the acyclicity of the resulting ground Bayesian network $B_\Phi(\mathcal{D}_S)$. Recall that the class \mathcal{S} must be defined via a (first-order) S -formula θ_S . As we are already employing the transitive closure operator in ψ'_Φ , we can also allow its use in θ_S , which is useful to express S -structures without cycles, for instance.

To check the consistency of Φ for a class \mathcal{S} , we cannot just check the validity of $\mathcal{C}'_{\Phi, \mathcal{S}} = \theta_S \rightarrow (\mathcal{B}_\Phi \rightarrow \psi'_\Phi)$, since θ_S specifies S -structures over D , while $\mathcal{B}_\Phi \rightarrow \psi'_\Phi$

presupposes the S -structure is given only over $D_S = \{d \in D \mid \mathcal{D} \models D_S(d)\} \subsetneq D$. To see the kind of problem that might occur, think of the class \mathcal{S} of all S -structures \mathcal{D} where each $d \in D$ is such that $s_i(d)$ holds, for some unary predefined relation $s_i \in S$. Consider an S -structure $\mathcal{D} \in \mathcal{S}$ ($\mathcal{D} \models \theta_S$), over a domain D . The formula \mathcal{B}_Φ cannot be satisfied by \mathcal{D} , for $D_S(x)$ must hold for all $x \in D$, because of the formulas in (8), so no $x \in D$ can represent ground formulas, due to the formulas in (5), contradicting the restrictions in (6) that require all ground atoms to be represented. Hence, this \mathcal{D} satisfies θ_S without encoding the ground Bayesian network, thus falsifying \mathcal{B}_Φ and satisfying $\mathcal{B}_\Phi \rightarrow \psi'_\Phi$, yielding the satisfaction of $\mathcal{C}'_{\Phi, \mathcal{S}}$. Consequently, $\mathcal{C}'_{\Phi, \mathcal{S}}$ is valid for this specific class \mathcal{S} , no matter how the relational Bayesian network Φ looks like. Nonetheless, it is not hard to think of a Φ that is trivially inconsistent for any class of S -structures, like $\Phi = \{F_r(x) = r(x)\}$, with $S = \emptyset$ and $R = \{r\}$, where the probability formula associated to the relation $r \in R$ is the indicator function $r(x)$, yielding a cyclic dependency relation \preceq .

In order to address the aforementioned issue, we need to adapt θ_S , constructing θ'_S to represent the class \mathcal{S} in the extended, bipartite domain $D = D_S \cup D_B$. The unary predicate $D_S(\cdot)$ is what delimits the portion of D that is dedicated to define the S -structure. Actually, we can define D_S as the set $\{x \in D \mid \mathcal{D} \models D_S(x)\} \subseteq D$. Therefore, we must construct a V -formula θ'_S such that the V -structure \mathcal{D} satisfies θ'_S iff the S -structure \mathcal{D}_S , formed by $D_S \subseteq D$ and the interpretation of the S relations, satisfies θ_S . That is, the S -formulas that hold in a S -structure $\mathcal{D}' \in \mathcal{S}$ must hold for the subset of a V -structure \mathcal{D} defined over the part of its domain that satisfies $D_S(\cdot)$. This can be performed by inserting *guards* in the quantifiers inside θ_S .

Definition 4 Given a (closed) S -formula θ_S , θ'_S is the formula resulting from applying the following substitutions to θ_S :

- Replace each $\exists x \varphi(x)$ in θ_S by $\exists x D_S(x) \wedge \varphi(x)$;
- Replace each $\forall x \varphi(x)$ in θ_S by $\forall x D_S(x) \rightarrow \varphi(x)$.

Finally, we can define the formula that encodes the consistency of a relational Bayesian network Φ for a class of S -structures \mathcal{S} :

Definition 5 For disjoint sets of relations S and R , a given relational Bayesian network Φ and a class of S -structures defined by θ_S , $\mathcal{C}_{\Phi, \mathcal{S}} = \theta'_S \rightarrow (\mathcal{B}_\Phi \rightarrow \psi'_\Phi)$.

The main result of this work is the translation of the consistency-checking problem to the validity of a formula from the transitive closure logic:

Theorem 1 For disjoint sets of relations S and R , a given relational Bayesian network Φ and a class of S -structures \mathcal{S} defined by θ_S , Φ is consistent for \mathcal{S} iff $\mathcal{C}_{\Phi, \mathcal{S}}$ is valid.

As first-order logic in general is already well-known to be undecidable, adding a transitive closure operator clearly does not make things easier. The first-order formula θ_S (and thus θ'_S) might already be undecidable, so we assume it is decidable. Ideally, we would like to show cases for which the validity of $\mathcal{C}_{\Phi, \mathcal{S}}$ would also be decidable. Nevertheless, even restricting the relations in R to be unary, besides assuming a decidable θ_S , we could not show the decidability of the resulting problem, although there are some decidable fragments of first-order logic with transitive closure operators – e.g. [Alechina and Immerman 2000, Ganzinger et al. 1999]. Conversely, we did not manage to prove general undecidability. Hence, the question is still open for investigations.

7. Conclusion and Future Work

In this work, we studied the problem of checking the consistency of relational Bayesian networks for a given class of input structures. We used first-order logic to encode the output ground Bayesian network into a first-order structure. Then we employed a transitive closure operator to express the acyclicity needed for consistency, finally reducing the consistency-check problem to that of deciding the validity of a logical formula.

We conjecture that Jaeger’s original proposal to the format of the formula encoding the consistency of a relational Bayesian network Φ for a class \mathcal{S} , $\theta_{\mathcal{S}} \rightarrow \psi_{\Phi}$, cannot be followed as stated. As we argued, the possible number of tuples built from a domain typically outnumbers its size, so that there is no straightforward way to encode the ground Bayesian network, whose nodes are ground atoms, into the input \mathcal{S} -structure. Therefore, it is hard to conceive of a method that translates the acyclicity of the ground Bayesian network into a formula ψ_{Φ} to be evaluated over an input structure in the class \mathcal{S} (satisfying $\theta_{\mathcal{S}}$).

Future work includes searching for decidable instances of the formula encoding the consistency of a relational Bayesian network for a class of input structures and exploring new applications for the logic techniques here developed.

References

- Alechina, N. and Immerman, N. (2000). Reachability logic: An efficient fragment of transitive closure logic. *Logic Journal of IGPL*, 8(3):325–337.
- Ganzinger, H., Meyer, C., and Veanes, M. (1999). The two-variable guarded fragment with transitive relations. In *Logic in Computer Science, 1999. Proceedings. 14th Symposium on*, pages 24–34. IEEE.
- Getoor, L. and Taskar, B. (2007). *Introduction to statistical relational learning*. MIT press.
- Jaeger, M. (1997). Relational bayesian networks. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 266–273. Morgan Kaufmann Publishers Inc.
- Jaeger, M. (2002). Relational bayesian networks: a survey. *Electronic Transactions in Artificial Intelligence*, 6:60.
- Jaeger, M. (2008). Model-theoretic expressivity analysis. *Lecture Notes in Computer Science*, 4911:325–340.
- Koller, D. (1999). Probabilistic relational models. In *International Conference on Inductive Logic Programming*, pages 3–13. Springer.
- Poole, D. (1993). Probabilistic horn abduction and bayesian networks. *Artificial intelligence*, 64(1):81–129.
- Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine learning*, 62(1):107–136.
- Sato, T. (1995). A statistical learning method for logic programs with distribution semantics. In *Proceedings of the 12th International Conference on Logic Programming (ICLP’95)*, pages 715–729. Citeseer.

Aprendizado Profundo Aplicado na Classificação de Microalgas

Iago Corrêa¹, Paulo Drews-Jr¹, Márcio S. de Souza², Virgínia Maria Tavano²

¹Centro de Ciências Computacionais– Universidade Federal do Rio Grande (FURG)
Rio Grande – RS – Brazil

²Instituto de Oceanografia– Universidade Federal do Rio Grande (FURG)
Rio Grande – RS – Brazil

{iago.correa,paulodrews}@furg.br, souzamsd@gmail.com, docvmtg@furg.br

Abstract. *Microalgae are unicellular organisms that have physical characteristics such as size, shape and other distinct ultrastructure features. Classifying them manually may require great effort from experts since thousands of microalgae can be found in a small sample of water. Furthermore, the manual classification is a non-trivial operation. We proposed a deep learning technique to solve the problem. The obtained results show an improvement in the classification quality in a end-to-end solution.*

Resumo. *Microalgas são organismos unicelulares que apresentam características físicas como tamanho, forma e outras ultra-estruturas específicas de cada grupo. Classificá-las manualmente pode exigir grande esforço de especialista, pois milhares de microalgas podem ser encontradas em uma pequena amostra de água. Além disso, a classificação manual é uma operação não trivial. Neste trabalho, é proposto o uso de técnicas de aprendizado profundo para resolver o problema. Os resultados obtidos mostram melhora na qualidade de classificação em uma solução fim-a-fim.*

1. Introdução

Microalgas são um grande grupo composto de protistas eucarióticos fotoautotróficos e cianobactérias procarióticas. Elas representam uma porção significativa de um grupo chamado fitoplâncton. As microalgas têm variados tamanhos, estruturas e formas. A classificação desses organismos é de grande valor nos estudos biológicos e oceanológicos, já que a descrição das espécies de microalgas presentes em um certo momento e lugar é importante para o entendimento a respeito de como a energia é transferida da base da cadeia alimentar para os níveis tróficos mais altos [Cullen et al. 2002]. Baseado neste conhecimento, torna-se possível entender as mudanças nos estoques de peixes e no ciclo do carbono de um dado ambiente. Um trabalho mais recente e com dados decadais, que reuniu pesquisadores de 80 nações, para o mapeamento global da biomassa bentônica e sua relação com a produção/biomassa fitoplanctônica demonstra a importância do estudo desses organismos [Costello et al. 2010].

As microalgas são classificadas em uma grande quantidade de grupos levando em consideração diferentes características, como por exemplo, a composição de pigmentos ou diversidade de ultraestruturas [Singh and Saxena 2015]. Dessa forma, o grupo apresenta

grande variedade morfológica como células arredondadas, ovais cilíndricas, fusiformes, além de projeções como por exemplo espinhos ou cílios. Além disso, a classificação do fitoplâncton também pode se dar pelo tamanho do organismo: picoplâncton (0.2-2 μm), nanoplâncton (2-20 μm) e micropoplâncton (20-200 μm).

Estudos da composição específica, estrutura de tamanho e biomassa a respeito das comunidades de fitoplâncton vêm sendo desenvolvidos através de métodos clássicos de microscopia ótica [Hamilton et al. 2001]. Através dessa abordagem há uma demanda muito grande de tempo para a realização de uma análise completa, pois é necessário a realização da manipulação manual por parte de um observador em amostras pequenas de água que podem apresentar muitos indivíduos com grande diversidade.

Com o objetivo de se obter informações de forma mais rápida sobre o ambiente aquático, o uso de analisadores de partículas tem sido uma ferramenta importante, pois esses analisadores possibilitam uma obtenção mais rápida e eficiente de informações sobre densidade, composição e morfometria dos organismos fitoplanctônicos. Tal equipamento automatizado é composto por um sistema ótico capaz de distinguir microalgas das demais partículas presentes na amostra e capturar imagens das mesmas, apresentando ainda um software que auxilia na classificação e visualização das células.

O analisador de partículas adotado neste trabalho é o FlowCAM [Jakobsen and Carstensen 2011]. Ele é capaz de obter informação a respeito de micro-organismos presentes em amostras de água, sendo assim uma das contribuições deste trabalho é a criação e a manutenção de um grande banco de dados de microalgas com mais de 29.000 amostras divididas em 24 classes, sendo que 19 dessas classes apresentam um número de organismos suficiente para a realização da classificação supervisionada através do aprendizado profundo.

Uma abordagem para a classificação dos dados extraídos dos analisadores de partículas seria de grande benefício para as pesquisas na área. No entanto, algumas questões importantes estão presentes nesta tarefa; as imagens extraídas do FlowCAM apresentam baixa resolução e neste tipo de dados está presente um desbalanço natural das classes. Além disso, outra questão importante é que os dados utilizados são classificados por diferentes especialistas que geralmente podem adotar diferentes critérios ao longo do tempo. Esses fatos tornam a classificação um problema cuja taxa de acerto atingida por humanos permanece entre 67% e 83% de acordo com [Culverhouse et al. 2003]. Portanto, o presente trabalho propõe a classificação supervisionada através de um modelo de redes neurais convolucionais para superar essas questões.

O modelo proposto neste trabalho consiste em uma rede neural convolucional. No melhor da busca no estado da arte, o presente trabalho é o primeiro a aplicar esse tipo de técnica na classificação de microalgas. As redes neurais convolucionais emergem como estado da arte em termos de acurácia em diferentes tarefas, em especial, apresenta bons resultados na tarefa de classificação de imagens [Szegedy et al. 2015], tornando-o de grande interesse para a classificação das microalgas com base nas imagens extraídas do FlowCAM.

Este artigo é organizado da forma que segue. A Seção 2 discutirá brevemente trabalhos anteriores sobre a classificação de microalgas ou outros organismos de interesse. Na Seção 3 é descrita a metodologia proposta. A Seção 4 traz os resultados obtidos usando

o banco de dados de microalgas proposto. Finalmente, conclusões e trabalhos futuros são apresentados na Seção 5.

2. Trabalhos Relacionados

Na literatura podem ser encontrados diferentes trabalhos que envolvem a tarefa classificatória através de aprendizado profundo, porém nenhum desses consiste exatamente na classificação do mesmo grupo taxonômico de microalgas com base em imagens extraídas de um analisador de partículas. Tal fato se deve a dificuldade de se obter dados classificados suficientes para o aprendizado desse tipo de rede. Além disso, o tamanho reduzido que permite apenas fornecer um pequeno número de características classificáveis faz do problema ainda mais desafiador, limitando as abordagens da literatura.

Diferentes abordagens de classificação baseada na extração prévia de características, por exemplo diâmetro, volume ou razão de aspecto dos organismos, foram avaliadas anteriormente. Alguns trabalhos foram realizados acerca da classificação de zooplâncton, que apesar de serem organismos maiores e com mais características observáveis, compartilham grosseiramente de certos aspectos com o fitoplâncton. [Blaschko et al. 2005], [Xu et al. 2010] e [Sosik and Olson 2007] conseguiram atingir resultados satisfatórios na classificação de plâncton sem utilizar de aprendizado profundo e utilizando características previamente extraídas e selecionadas dos organismos.

No melhor do nosso conhecimento, não há outros trabalhos acerca de métodos de classificação de microalgas baseado em imagens, sem qualquer pré-processamento para análise e seleção de características. [Dai et al. 2016] trabalharam na classificação de zooplâncton baseado em imagens em escala de cinza capturadas pelo sistema ZooScan [Gorsky et al. 2010]. Uma arquitetura de aprendizado profundo associada a um aumento de dados através de amostras artificiais foi proposta para a classificação desse zooplâncton de forma automática. Nenhuma característica foi previamente selecionada ou extraída dos organismos e, assim, o problema foi superado com uma acurácia de 93.7% para 13 classes com imagens de 256×256 pixels. Apesar do zooplâncton ser composto de organismos maiores, o problema da baixa resolução de imagens e desbalanço natural das classes também está presente.

Por fim, estudando a classificação dos mesmos organismos [Drews-Jr et al. 2012, Drews-Jr et al. 2013] obtiveram resultados significativos de 92% de acurácia na classificação de microalgas utilizando modelos de misturas Gaussianas junto de aprendizado semi-supervisionado e aprendizado ativo em um banco de apenas quatro classes. Apesar do bom resultado obtido, o trabalho não lidava com o desbalanço, o tamanho do banco de dados utilizado era reduzido e um especialista deveria participar ativamente no processo de classificação. O reduzido tamanho do banco de dados facilita bastante a tarefa, não permitindo uma comparação justa entre o presente trabalho e o trabalho anterior. [Corrêa et al. 2016] buscaram melhores resultados na classificação levando em consideração o desbalanço de classes e atingiram valores altos de acurácia através da utilização de modelos de classificação populares com técnicas de amostragem em um conjunto de 19 classes distintas. Mesmo com valores significativos de acurácia, a abordagem proposta depende intensamente de técnicas de amostragem e do pré-processamento das imagens para obtenção de características visuais.

3. Metodologia

Este trabalho apresenta um modelo de rede neural convolucional, do inglês CNN (*Convolutional Neural Network*), como abordagem para classificar de forma eficiente as imagens extraídas das microalgas. Então, inicialmente é apresentada uma introdução de como esses modelos funcionam e do porquê de obterem características significantes das imagens. Então são apresentados a estrutura e processamentos associados ao banco de dados utilizado na classificação.

3.1. Redes Neurais Convolucionais

Para classificar de maneira adequada as imagens das microalgas em baixa qualidade e assim com limitada quantidade de informação, se torna necessário um modelo poderoso capaz de aprender com os dados disponíveis. Nesse sentido, têm-se os modelos de aprendizado profundo. Os modelos têm estado em evidência devido à sua grande capacidade de capturar características visuais da hierarquia dos dados. Esses modelos são aplicados em muitas aplicações [Glorot et al. 2011] [Kalchbrenner et al. 2014] atingindo resultados promissores. Tais resultados são possíveis devido a um alto nível de abstração para dados visuais segundo a utilização de uma estrutura profunda, ou seja, uma quantidade massiva de transformações não lineares e parâmetros que são ajustados.

Dentre os modelos de aprendizado profundo, as CNNs têm se tornado o padrão para a classificação de entradas visuais [Jarrett et al. 2009]. Esses modelos são redes neurais artificiais de alimentação progressiva bio-inspiradas no córtex visual e apresentam uma arquitetura que é composta de camadas que pegam um dado volume de entrada e o transformam em um novo volume de saída.

Essas camadas para a classificação de imagens são geralmente compostas de repetições de conjuntos de convoluções, funções não-lineares e camadas de *pooling*, seguidos de uma camada completamente conectada com a camada de ativação. A camada de *pooling* consiste na realização de um *max pooling*, ou seja, recombinações lineares que diminuem a dimensionalidade, permitindo uma convergência mais rápida devido à seleção de características invariantes e assim aumentando a capacidade de generalização [Nagi et al. 2011]. A função não linear mais utilizada é o ReLU (*Rectified Linear Unit*), pois seu desempenho é melhor na maioria das situações [Krizhevsky et al. 2012].

3.2. Banco de Dados

Os dados são extraídos de amostras de água do Oceano Atlântico Sul (latitude 35°S) utilizando o analisador de partículas FlowCAM. O equipamento fornece 26 características [Jakobsen and Carstensen 2011] e imagens de cada organismo-alvo analisado. Análises anteriores neste tipo de dados utilizaram uma seleção de algumas dessas propriedades para a classificação [Drews-Jr et al. 2012, Drews-Jr et al. 2013, Corrêa et al. 2016], mas para este trabalho nenhuma característica previamente extraída é de interesse, apenas as imagens são utilizadas para o propósito classificatório.

O banco de dados de microalgas é composto por 19 classes com 29.449 imagens. As classes presentes no banco de dados e seus respectivos números de amostras são: Bacillariophycidae-1 (753), Nannoplankton $< 5\mu\text{m}$ (569), Nannoplankton 5-10 μm (20916), Prorocentrales (211), Spirotrichea (208), Peridinales (189), Gymnodinales (137), Cochlodinium (13), Ellipsoidal Microplankton (171), Spherical Microplankton

(215), Ellipsoidal Nannoplankton 10-20 μm (621), Spherical Nannoplankton 10-20 μm (3538), Pseudo-nitzschia (703), Bacillariophycidae-2 (261), Chaetocerotales (548), Coscinodiscophyceae (79), Corethron (39), Rhizosoleniales (187), and Naviculales (91). Na Figura 1 é apresentada uma amostra de cada uma das classes.

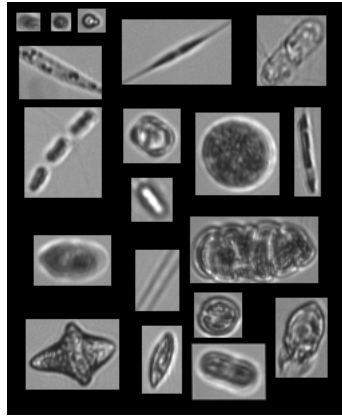


Figura 1. Uma amostra de cada classe do banco de dados utilizado. As microalgas são adquiridas utilizando o analisador de partículas FlowCAM.

Para o pré-processamento das imagens que compõem o banco de dados o primeiro passo adotado foi o redimensionamento das imagens, pois apresentam ampla variação de tamanho. Como as imagens originais não apresentam alta resolução, o tamanho adotado foi de 64×64 pixels com a proporção preservada. Depois disso, é aplicada uma normalização para tornar a média zero e fazer com que o desvio padrão seja unitário. Outras técnicas poderiam ser aplicadas, no entanto o objetivo para este trabalho é atingir bons resultados com o menor pré-processamento possível.

3.3. Arquitetura da Rede

Diferentes arquiteturas foram testadas com o objetivo de classificar os organismos da melhor forma possível tendo como referência de validação a acurácia e uma função de erro. Esse erro é associado com a entropia cruzada entre duas distribuições de probabilidades, uma aproximada e outra verdadeira, ou seja, a distribuição de probabilidades prevista e a correta. Dessa forma, foram investigadas redes com diferentes profundidades e números de convoluções, além da influência de maior ou menor regularização.

3.3.1. Treino e validação

Como forma de avaliar apropriadamente os modelos é necessário que seja feita uma divisão do banco de dados em uma parte destinada ao treinamento do modelo e outra parte destinada à validação. Dessa forma, a acurácia obtida pelo modelo é avaliada somente com dados que não estiveram presentes na etapa de treino, evitando que se ocorra o *overfitting*, ou seja, evitando que o modelo se torne específico de mais sendo incapaz de generalizar o aprendizado para amostras novas. Para este trabalho optou-se por utilizar uma divisão de 70% dos dados para a etapa de treino e os 30% restantes para validação.

Além disso, durante a etapa de treino, o banco de dados de treino foi dividido em *mini-batches*, ou seja, divisões bem menores do banco como forma de otimizar o tempo de treinamento, sendo uma vantagem também para o uso de placas de vídeo [Seide et al. 2011]. Não houve qualquer tratamento para tornar esses *mini-batches* mais balanceados, sendo assim, eles mantêm a característica de desbalanço do banco de dados utilizado.

3.3.2. Número de camadas e as convoluções

O número de camadas, ou profundidade, pode ser considerado o fator mais importante de um modelo, pois conforme a rede se torna mais complexa este tende a aumentar. Porém, apesar da profundidade estar associada a melhores resultados, é necessário que haja mais dados para acompanhar o aumento e sendo assim não é qualquer modelo que poderá comportar um número grande de camadas [Hinton et al. 2006]. Além disso, o tamanho das convoluções ou *kernel*, bem como o número de convoluções também são importantes para o desempenho do modelo.

Modelos com diversas profundidades e tamanho de convoluções foram testados em busca de um melhor resultado. Devido ao pequeno tamanho e baixa qualidade das imagens dos organismos a serem classificados, convoluções muito grandes não obtiveram resultados relevantes, bem como modelos de profundidade muito grande. Tal fato se deve à quantidade de dados disponíveis em comparação ao número de classes a serem classificadas. Três arquiteturas que apresentaram os resultados mais relevantes se encontram na Tabela 1, a primeira linha identifica o número de camadas e o uso de regularização e nas demais linhas são especificadas a composição das camadas. Os resultados para esses modelos estão presentes na Tabela 2.

Também é importante mencionar que as camadas de ativação utilizam o ReLU como função de ativação. Além disso, foi utilizado o *Max-Pooling* com filtros 2×2 nas camadas de *pooling*.

Modelos maiores ou menores não estão dispostos na tabela devido aos baixos resultados e assim, é possível perceber que os dados geram resultados melhores com modelos que apresentam 8 camadas e com convoluções de 3×3 ou 7×7 . Para o modelo final optou-se para o uso de 8 camadas onde a primeira camada é composta por 16 convoluções de tamanho 7×7 , acompanhado de mais 4 camadas de convolução e 3 de redes completamente conectadas conforme mostrado no modelo final presente na figura 2.

4. Resultados Experimentais

Após definição de um modelo final conforme discutido na seção anterior, foram avaliadas como métrica de avaliação deste modelo a acurácia e o erro na validação do mesmo. Seus resultados foram de 71.12% de acurácia e um erro 4.67, mostrando assim uma boa chance de acertos mediante o desbalanço dos dados não ter sido diretamente tratado, pouco pré-processamento e nenhuma seleção prévia de características relevantes sobre os organismos classificados.

Tabela 1. Diferentes arquiteturas testadas para a classificação de microalgas.

6 layers	7 layers	8 layers
Input: 64x64 image		
Convolução 16:3x3	Convolução 16:7x7	Convolução 16:3x3
		Convolução 32:3x3
Max-pooling 2x2		
Convolução 64:3x3	Convolução 64:3x3	Convolução 64:3x3
	Convolução 128:3x3	Convolução 128:3x3
Max-pooling 2x2		
Convolução 256:3x3	Convolução 512:3x3	Convolução 256:3x3
Max-pooling 2x2		
FC 4000	FC 6400	FC 6400
FC 512		
Dropout		
FC 19		

Tabela 2. Resultados de classificação em microalgas para diferentes profundidades, convoluções e regularização.

	Acurácia	Erro	
	Validação	Treinamento	Validação
6 layers	0.6334	0.09	3.5183
7 layers	0.6387	0.1	3.5735
8 layers	0.6376	0.1015	3.5907

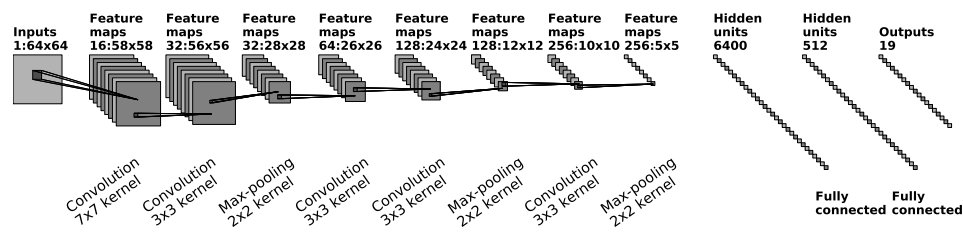
5. Conclusão e Trabalhos Futuros

Este trabalho apresenta o resultado da busca por um modelo de aprendizado profundo com redes neurais convolucionais para realizar a classificação de microalgas.

Além disso, um modelo final é obtido apresentando 8 camadas, sendo 5 convolucionais e 3 camadas completamente conectadas, além de apresentar resultados de 71.12% de acurácia com muito pouco pré-processamento e sem qualquer intervenção de especialista para selecionar características relevantes das microalgas que poderiam facilitar a classificação. Tal abordagem prova sua vantagem uma vez que apresenta uma taxa de acerto esperado similar a de um especialista humano [Culverhouse et al. 2003], é mais independente de pré-processamento de dados do que abordagens do estado da arte para a classificação desses organismos e não apresenta nenhuma abordagem direta ao desbalançamento dos dados.

Para trabalhos futuros é necessário continuar a expansão do banco de dados atual como forma de obter melhores resultados e de poder realizar a classificação das microalgas em mais classes que não constem ainda nos dados. Pretende-se utilizar *data aug-*

Figura 2. Rede neural convolucional para aprendizado profundo de microalgas.



mentation de modo a aplicar o banco de dados. Além disso, pretende-se aplicar o conceito de redes neurais adversárias como forma de melhorar os resultados com menores quantidades de dados classificados. Finalmente também é de interesse o trabalho para o compartilhamento dos dados tendo como objetivo seu uso ou possíveis colaborações.

Agradecimentos

Agradecemos ao CNPQ, FAPERGS e FURG por suportar parcialmente o desenvolvimento desta pesquisa. Além disso, também agradecemos a empresa NVIDIA por realizar a doação de uma Titan X Pascal utilizada nos treinos das redes de aprendizado profundo.

References

- Blaschko, M. B., Holness, G., Mattar, M. A., Lisin, D., Utgoff, P. E., Hanson, A. R., Schultz, H., and Riseman, E. M. (2005). Automatic in situ identification of plankton. In *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1*, volume 1, pages 79–86.
- Corrêa, I., Drews, P., d. Souza, M. S., and Tavano, V. M. (2016). Supervised microalgae classification in imbalanced dataset. In *5th BRACIS*, pages 49–54.
- Costello, M. J., Coll, M., Danovaro, R., Halpin, P., Ojaveer, H., and Miloslavich, P. (2010). A census of marine biodiversity knowledge, resources, and future challenges. *PLoS one*, 5(8).
- Cullen, J. J., Franks, P. J., Karl, D. M., and Longhurst, A. (2002). Physical influences on marine ecosystem dynamics. *The sea*, 12:297–336.
- Culverhouse, P., Williams, R., Reguera, B., Herry, V., and González-Gil, S. (2003). Do experts make mistakes? a comparison of human and machine identification of dinoflagellates. *Marine Ecology-progress Series*, 247:17–25.
- Dai, J., Wang, R., Zheng, H., Ji, G., and Qiao, X. (2016). Zooplanktonet: Deep convolutional network for zooplankton classification. In *OCEANS 2016*, pages 1–6.
- Drews-Jr, P., Colares, R., Machado, P., de Faria, M., Detoni, A., and Tavano, V. (2012). Aprendizado ativo e semi-supervisionado na classificação de microalgas (in portuguese). In *IX ENIA*.

- Drews-Jr, P., Colares, R., Machado, P., de Faria, M., Detoni, A., and Tavano, V. (2013). Microalgae classification using semi-supervised and active learning based on Gaussian mixture models. *Journal of the Brazilian Computer Society*, 19(4):411–422.
- Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In Getoor, L. and Scheffer, T., editors, *28th ICML*, pages 513–520, New York, NY, USA.
- Gorsky, G., Ohman, M. D., Picheral, M., Gasparini, S., Stemmann, L., Romagnan, J.-B., Cawood, A., Pesant, S., García-Comas, C., and Prejger, F. (2010). Digital zooplankton image analysis using the zooscan integrated system. *Journal of Plankton Research*, 32(3):285.
- Hamilton, P., Proulx, M., and Earle, C. (2001). Enumerating phytoplankton with an upright compound microscope using a modified settling chamber. *Hydrobiologia*, 444(1-3):171–175.
- Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554.
- Jakobsen, H. and Carstensen, J. (2011). FlowCAM: Sizing cells and understanding the impact of size distribution on biovolume of planktonic community structure. *Aquatic Microbial Ecology*, 65:75–87.
- Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? In *12th ICCV*, pages 2146–2153.
- Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *CoRR*, abs/1404.2188.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.
- Nagi, J., Ducatelle, F., Di Caro, G. A., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., and Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *ICSIPA 2011*, pages 342–347.
- Seide, F., Li, G., and Yu, D. (2011). Conversational speech transcription using context-dependent deep neural networks. In *INTERSPEECH 2011*.
- Singh, J. and Saxena, R. C. (2015). An introduction to microalgae. In *Handbook of Marine Microalgae*, pages 11–24. Elsevier Science.
- Sosik, H. M. and Olson, R. J. (2007). Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. *Limnology and Oceanography: Methods*, 5(204):e216.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.
- Xu, L., Jiang, T., Xie, J., and Zheng, S. (2010). Red tide algae classification using svm-snp and semi-supervised fcm. In *ICETC*, volume 1, pages V1–389.

Application of Bicluster Algorithms in Human Aging Database

Marta Dias Moreira Noronha¹, Caio Eduardo Ribeiro¹, Luis Enrique Zárate¹

¹Departamento de Informática
Pontifícia Universidade Católica de Minas Gerais (PUCMG)
Belo Horizonte – MG – Brasil

`martadmnoronha@gmail.com, caioedurib@gmail.com, zarate@pucminas.br`

Abstract. *The high dimensionality of databases hinders the application of distance metrics by clustering algorithms, reducing the overall cluster quality. In this study, we propose the application of biclustering algorithms in a database, derived from longitudinal database, for the extraction of meaningful clustering. This research aims to select biclusters of long-lived and short-lived individuals from the aging longitudinal database. The experiments performed showed good biclusterings for the long-lived class, including some overlapping attributes and registers, however only one bicluster has been found with over 60% of short-lived.*

1. Introduction

Currently, the Machine Learning community has been facing challenges where it is necessary to process great volumes of data simultaneously. Conventional algorithms and techniques such as clustering may be unable to present satisfactory results when there is a high dimensionality in the attributes of the dataset. Usually similarity measures such as Euclidean Distance tend to an average value, hindering the differentiation of registers, in high dimensional datasets [Beyer et al. 1999]. Several approaches have been proposed to deal with this issue, such as applying dimensionality reduction techniques, or weighting the more relevant attributes. Others apply partial clustering, e.g., in heterogeneous datasets it is possible to start the clustering considering numerical value attributes, followed by nominal value ones in another step, considering the groups already found. Another alternative is to apply biclustering techniques over a dataset, because they allow the grouping of registers regarding a subset of attributes and/or the grouping of attributes considering a subset of registers [Cheng and Church 2000]. Biclustering algorithms types can be differentiated from each other by the type of bicluster generated, which can have a similarity over every element contained in the bicluster or not, be positioned differently in the dataset, and also be generated through different search heuristics [Madeira and Oliveira 2004]. After the biclustering process, it is possible to apply clustering techniques over the results, considering the attribute subset of the dataset, to further investigate the results.

Longitudinal databases are one type of high dimensional database. Such databases are usually generated by social and health studies where a sample of individuals is interviewed throughout a span of years, in time periods called waves. Longitudinal databases can reach about 50,000 registers and have thousands of attributes, because of the comprehensiveness of the studies they come from. In longitudinal databases the registers have a temporal aspect, because they represent the same individual over each time unit, and are difficult to differentiate due to the high number of attributes considered. The Knowledge

Discovery in Databases (KDD) process, when applied to such databases, should take in consideration these aspects to avoid loss of time-related information [Ribeiro et al. 2017].

In this paper, as a case study, we considered a longitudinal database of an ageing study. Several studies have been made to understand the process of human ageing, in order to discover knowledge regarding the factors that influence longevity, so that the quality of life of the elderly population can increase continuously. It is estimated that the rate of people aged 60 and over currently represents about 12.3% of the world population and, by 2050 this rate will increase to about 21.5%, and continue to rise, having a huge social and financial impact. Some databases from longitudinal studies of ageing worth mention are the Survey of Health Ageing and Retirement (SHARE), which is currently in its 5th wave and has nearly 110.000 respondents from 21 different countries, Health and Retirement Study (HRS), currently in its 13th wave with about 26.000 respondents in the USA, the Chinese Longitudinal Health Longevity Survey (CLHLS), in its 5th wave with about 19.000 Chinese respondents, and finally the English Longitudinal Study of Ageing (ELSA ¹) [Ribeiro et al. 2017].

The ELSA is a study conducted in the United Kingdom, currently in its 7th wave, and it has about 12.000 respondents, who answer questionnaires applied by professional biannually, since 2002. Each wave has attributes relating to eight dimensions that can influence human ageing in different manners, those being: quality of life, physical health, mental health, genetic inheritance, social, personal, economic and location. After the pre-processing realized in a previous work, the dataset has 1.333 registers and 129 attributes, and 48 of those are binary attributes, which will be inputted to biclustering algorithms in this work, as a pre-grouping process to characterize the profiles of long-lived and short-lived individuals of the database. This derived database from ELSA has not the temporal aspects, but it is difficult separate the classes in this database with 48 attributes. [Ribeiro et al. 2017].

According to [Ribeiro 2017], the high dimensionality of a dataset can affect the quality of the results because the distances between the registers become closer as attributes are added. And, as can be observed in [Ribeiro 2017], in a dataset with 50 attributes, it is already difficult to obtain measures of similarity or dissimilarity that separate the classes well. Therefore, it is necessary to pre-process the dataset for obtain dimensionality reduction and, on a longitudinal database, care must be taken to avoid loss of information during the KDD process.

Our main objective is to identify clusters to profile long-lived and short-lived individuals, considering the binary attributes from the database derived from ELSA longitudinal database, using and comparing different biclustering algorithms from the literature. We intend to discover register biclusters (from the long-lived and short-lived class), that are similar regarding subsets of attributes. As 82 years is the average lifespan of an UK citizen, we considered as long-lived every ELSA respondent whose age reached beyond that during the study, and took their registers as representatives of the long-lived class, while considering short-lived those who were deceased prior to reaching 82 years old, for non-accidental reasons.

This paper is organized as follows. In Section 2, we present the main heuristics

¹Site: <http://www.elsa-project.ac.uk/>

used by the main biclustering algorithms, the bicluster types and structures, and evaluation forms. Next, Section 3 contains the materials and methods used in the biclustering experiment performed. Section 4 presents the procedures and result analysis, followed by a Section 5, which contains our conclusions and future work recommendations.

2. Biclustering

Understanding the basic concepts of the different bicluster types, structures and heuristics is necessary for the definition of evaluation methods and to validate our results. It is important to emphasize that biclusters might be overlapped, which makes conventional evaluation methods such as Silhouette Index unfeasible [Zhao et al. 2012].

2.1. Types of Biclusters

Biclusters can have the following classifications, according to [Madeira and Oliveira 2004, Zhao et al. 2012, Saber and Elloumi 2014]:

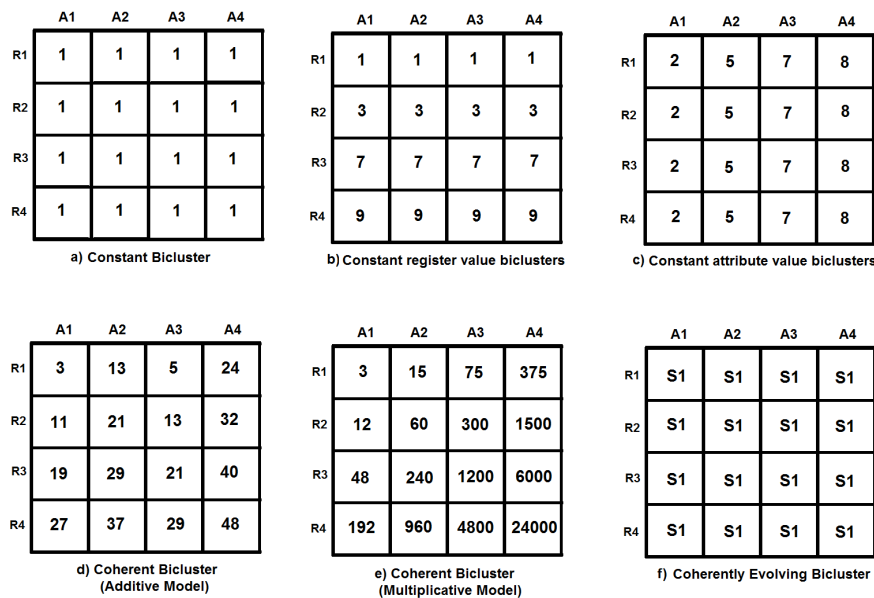


Figure 1. Types of Biclusters. Source: Adapted from [Madeira and Oliveira 2004]

a) Constant Bicluster: Each element a_{ij} in the bicluster has the same value (Figure 1.a). If there are additive or multiplicative noise applied exclusively over registers or attributes in the database, it is possible to encounter constant register value biclusters (Figure 1.b), or constant attribute value biclusters (Figure 1.c).

b) Coherent Bicluster: A bicluster encountered in a database with additive (Figure 1.d) or multiplicative (Figure 1.e) noise values over its registers and/or attributes.

c) Coherently Evolving Bicluster: Presents a symbolic (Figure 1.f) or numeric pattern on its registers, regardless of the average value of each ordered pair (i,j) in the bicluster. This pattern may be an ascending order of the values in every attribute in the bicluster, e.g., attribute1 < attribute3 < attribute4 < attribute2.

2.2. Bicluster Structures

According to [Madeira and Oliveira 2004] and [Saber and Elloumi 2014], bicluster structures differentiate regarding allowing overlaps or not. A dataset may have only a single

bicluster (Figure 2.a). Each register and attribute may only be assigned to a single bicluster (Figure 2.b). Every bicluster in the dataset may have a similar structure, such as a checkers board, not allowing overlaps of either attributes or registers (Figure 2.c). A structure of register exclusivity (Figure 2.d) requires that each register is assigned to a single bicluster, but permits that attributes overlap. Attribute exclusive bicluster revert that logic (Figure 2.e), not allowing attribute overlap but allowing the same registers to be part of different biclusters. Non-overlapping biclusters that can be modelled with a tree structure (Figure 2.f), require that each attribute and register is assigned to at least one bicluster. The non-exclusive and non-overlapping bicluster (Figure 2.g) allows its registers and attributes to belong to more than one bicluster as long as there is no bicluster overlapping. A hierarchical structure bicluster (Figure 2.h) allows an overlapping of registers and attributes, even permitting that they are not assigned to any bicluster, and allows a hierarchy to be established between biclusters in the dataset. Finally, it is possible to have arbitrarily positioned biclusters (Figure 2.i), which does not comprise any requirements [Madeira and Oliveira 2004, Saber and Elloumi 2014].

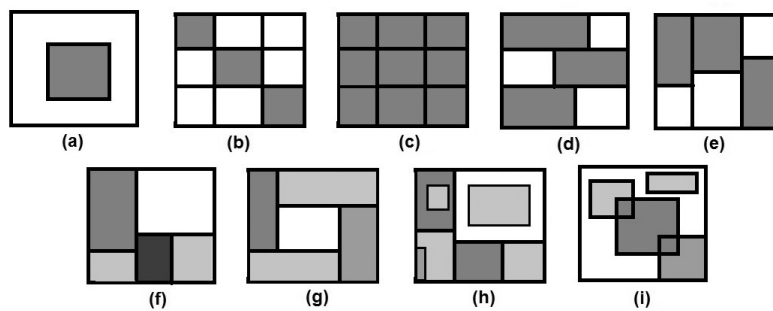


Figure 2. Biclusters Structures. Source: [Madeira and Oliveira 2004]

2.3. Heuristics for Biclustering Algorithms

The biclustering heuristics are developed to find sub-optimal solutions, given that the n -dimensional partition problem is in the NP-Hard complexity class [Madeira and Oliveira 2004]. The heuristics are based upon:

- a) Iterative Greedy Search: Finds biclusters using a function of maximization of local gains, frequently finding suitable biclusters even when wrong decisions are made;
- b) Divide and Conquer: A fast data partition process, which might lose good biclusters when subsets of data are generated;
- c) Restriction of the dataset dimension and Exhaustive enumeration: This heuristic performs a search for the very best biclusters in the dataset, with a high computational cost. In [Veroneze et al. 2017], the authors propose methods of enumerating biclusters based on Formal Concept Analysis (FCA);
- d) Statistical Model: Identifies distribution parameters through a minimization function that allows the discovery of biclusters;
- e) Separately clustering of attributes and registers: This technique is used in combination to a result merging method to find biclusters in the dataset.

2.4. Biclustering Evaluation

A bicluster may be evaluated internally, or have its relation to other biclusters assessed. When evaluating a bicluster internally, we can obtain its coherence degree, for its registers and attributes simultaneously or separately, which would require a partial result combination to get the bicluster result. When evaluating inter-biclusters, it is possible to access the accuracy of the algorithm by comparing its ability to find true biclusters in a synthetic dataset, using another algorithm as reference, as was done in [Prelić et al. 2006], where the authors used Bimax as a reference algorithm in the comparison, because it made an exhaustive search for every optimal bicluster in the dataset [Saber and Elloumi 2014].

2.5. Biclustering Validation

After obtaining a bicluster with an algorithm, the results may be statistically validated through measures such as coverage, separation, and bicluster connectivity. We can highlight three coverage measures [Saber and Elloumi 2014] (note: elements are ordered pairs of registers and attributes):

$$\text{Database Coverage} = \text{elements found in the biclusters} / \text{elements in the database} \quad (1)$$

$$\text{Register Coverage} = \text{registers found in the biclusters} / \text{registers in the database} \quad (2)$$

$$\text{Attribute Coverage} = \text{attributes found in the biclusters} / \text{attributes in the database} \quad (3)$$

There are also domain specific validation techniques. In Bioinformatics researches, e.g., biclusters are frequently validated through Gene Ontology, used to access the algorithm's capacity to extract biclusters that present a biological pattern [Saber and Elloumi 2014].

If the registers are labelled, it is possible to use the hypergeometric distribution function p-value (Equation 4) to measure bicluster quality. Considering that the dataset has n classes, [Zhao et al. 2012] define as $|C_i|$ the majoritarian class in the bicluster, as M the total of registers in the dataset, as r the number of hits considering the majoritarian class of the bicluster, and as b the total of registers of the majoritarian class in the dataset. This measure considers the probability of including a register from the majoritarian class by chance.

$$p\text{-value}(B) = \frac{\binom{|C_i|}{r} \binom{M-|C_i|}{b-r}}{\binom{M}{b}} \quad (4)$$

We considered that a p-value lower than 0.05 indicates that the bicluster presents statistical representativeness and significance [Ayadi et al. 2009].

2.6. Biclustering Algorithms

Researches pointed out new approaches of biclustering algorithms for continuous or categorical databases, for instance, BiBinCons[Saber and Elloumi 2015], BiBinAlter[Saber and Elloumi 2015] and HBLCoClust[de França 2016]. However, in this study, we considered the 10 most cited biclustering algorithms, according to our research, in the CAPES (Comissão de Aperfeiçoamento de Pessoal do Nível Superior) website. But it was necessary to do further research on biclustering algorithms that receive binary inputs, because only one of the 10 we found was ideal for this type of data.

Table 1 summarizes the main characteristics of the most cited algorithms, which were further investigated to decide which would be used in our ELSA biclustering experiment.

As we stated before, this biclustering of the database is one pre-grouping step in a larger Knowledge Discovery in Databases process being in development, which ultimately aims to profile long-lived and short-lived ELSA respondents. In this work, we hope to find attribute subsets that best separate registers from both classes. In addition to that, biclustering techniques can be useful in rare pattern discovery problems, a category in which the ELSA database fits, for its registers are similar (mainly because of its high dimensionality), which hinders the differentiation of its classes.

Table 1. Biclustering Algorithms. Source: Adapted from Madeira and Oliveira (2004), Charrad and Ben Ahmed (2011) and Higa et al. (2010)

Algorithm [Reference]	Type of Bicluster	Structure	Heuristic Approach
δ -bicluster [Cheng and Church 2000]	Coherent	Randomly positioned overlap	Iterative Greedy Search
Bimax [Prélic et al. 2006]	Coherent	Hierarchy overlap	Divide and conquer
Plaid [lazzeroni and Owen 2002]	Coherent	Randomly positioned overlap	Distribution parameters
ISA [Bergmann et al. 2003]	Coherent	Randomly positioned overlap	Iterative Greedy Search
BSGP [Dhillon 2001]	Coherent	Non-exclusive without overlap	Exhaustive enumeration
SAMBA [Tanay et al. 2002]	Coherently evolving	Randomly positioned overlap	Exhaustive enumeration
OPSM [Ben-Dor et al. 2003]	Coherently evolving	Simple or Randomly positioned overlap	Iterative Greedy Search
Spectral [Kluger et al. 2003]	Coherent	Overlap with grid structure	Iterative Greedy Search
FLOC [Yang et al. 2005]	Coherent	Randomly positioned overlap	Iterative Greedy Search
xMotifs [Murali and Kasif 2003]	Coherently evolving	Simple or Randomly positioned overlap	Iterative Greedy Search
Bibit [Rodriguez-Baena et al. 2011]	Coherent	Randomly positioned overlap	Bit pattern search using encoding

The structure that allows the occurrence of overlap between biclusters and the acceptance of input binary data were the primary requirements for the selection of the algorithms applied in the data set. The algorithms that process continuous data inputs were selected because they use thresholds, to convert the base to binary data, or some binary parameter that, through some function, associates a data in a bicluster. The last criterion of choice was based on the availability of the algorithm for execution.

From the filtering of the algorithms, through the chosen criteria, the BiMax [Prelić et al. 2006], Bibit [Rodriguez-Baena et al. 2011], ISA [Bergmann et al. 2003] and Plaid [Lazzeroni and Owen 2002] algorithms were selected for application on binary data from the ELSA base.

3. Methodology

3.1. Materials - Dataset

In this paper, we utilized only the binary attributes of the pre-processed version of ELSA generated in a previous work [Ribeiro and Zárata 2017]. Preprocessing was necessary to eliminate outliers and analyze data that were not present in all waves, because a longitudinal dataset should contain the same registers and attributes in all waves. Other steps, such as attribute selection, data transformations, and other KDD processes to prepare the longitudinal data, can be found in [Ribeiro 2017]. From this process, two datasets derived from ELSA have emerged. One of which has its temporal aspects preserved and, the other, contains data about people who achieved longevity and data about those who died before reaching longevity. We use, in this work, the second dataset which contains 1.333 registers and its 48 binary attributes.

The binary attributes were, previously, separated in blocks that describe it as: Gender, Frequency of physical activities, Physical limitations, Medical history, Medication use, Pain and intensity, Depression symptoms questionnaire, Severe cognitive problems, Limitations caused by health problems, Mobility test results, Recent formal education, General family / social structure, Economic Situation: house / vehicle, Economic Situation: career and Economic Situation: Current economic state.

As stated before, there is a class imbalance, with 1.091 of the registers in the database belonging to the long-lived class, in contrast to only 242 of the short-lived, so we would expect a difficulty in finding biclusters from the minority.

3.2. Materials - Biclustering Algorithms

Bimax [Prelić et al. 2006] is a divide and conquer algorithm able to find every optimal bicluster in feasible time. The input parameters are the minimal number of registers (minr) and attributes (minc) that each bicluster may have, and the maximum (number) of biclusters that should be found in the dataset.

Bibit [Rodriguez-Baena et al. 2011] is a fast bit processing technique that uses coding to search for patterns. It is relatively more efficiently than Bimax, even though their results are similar, because it is unaffected by the shape or density of the input data. The input parameters are the minimal of registers (minr) and attributes (minc) in an acceptable bicluster.

The ISA [Bergmann et al. 2003] algorithm differentiates from the more common SVD (Singular Value Decomposition) algorithm by using two threshold, one for registers (thr.row) and another for attributes (thr.col). These threshold are utilized to calculate the average and standard deviation of the vector, and keep only the registers and attributes that are between acceptable values. The biclustering process is applied firstly over the continuous input data, followed by the binary, with the thresholds as input parameters.

The Plaid Model [Lazzeroni and Owen 2002] algorithm receives continuous valued data as input, and utilized a linear function (fit.model), based in the δ -bicluster algorithm, to find coherent biclusters, relaxing its constraints through a parameter of association of binary groups for registers, and another for attributes, besides adding a residual error value in each layer of its process. This algorithm requires that the user defines if it should perform the biclustering over its registers, attributes, or both. It is also possible to change the threshold of trimming registers (row.release) and attributes (col.release) in the layers, depending on the homogeneity of said register/attribute. The initial number of iterations to find the initial values (iter.startup), the number of iterations in each layer (iter.layer), the degree of freedom in the background layer (background.df), the added layers adjustments (back.fit), the maximum of layers to be added (max.layers), and “shuffle” parameter, which evaluates statistical significance before adding a layer may also be modified, to configure the algorithm.

3.3. Tools

The experiments were performed using the RStudio² framework, which implements codes in the open-source statistical code language R, frequently used in research studies. The packages used for the implementation for the biclustering algorithms in the framework are: biclust³, isa2⁴ and BibitR⁵. We also created simple C programs to verify the quantity of registers from each class in the biclusters generated, for the result validation process, when we determined the measures of coverage and the p-value of the biclusters, since these metrics were not implemented in R.

4. Experimental Procedures and Results

We applied selected algorithms in the section 3.2 on the database with 1.333 registers and 48 attributes. The amount of discovered biclusters by each algorithm are presented in Table 2, as are the input parameters (acronyms described in the Section 3.2) used in that execution.

Table 2. Input Parameters and Biclusters Found. Source: Search data

Algorithm	Biclusters found	Parameters
Bibit	368447	minr = 40, minc = 5
Bimax ¹	299.900	minr = 200, minc = 30, number = 300000 ⁶
Bimax ²	58	minr = 300, minc = 30, number = 1500
Bimax ³	13	minr = 310, minc = 30, number = 1500
Bimax ⁴	4	minr = 315, minc = 30, number = 1500
Bimax ⁵	2	minr = 320, minc = 30, number = 1500
ISA	15	thr.row = 1, thr.col = 1
Plaid	7	cluster = "b", background.df = 4, row.release = 0.7, col.release = 0.7, back.fit = 0, max.layers = 20, iter.startup = 20, iter.layer = 20

²Site: <https://www.rstudio.com/>

³Site: <https://cran.r-project.org/web/packages/biclust/index.html>

⁴Site: <https://cran.r-project.org/web/packages/isa2/index.html>

⁵Site: <https://cran.r-project.org/web/packages/BiBitR/index.html>

The results of the validation measures (as in 1, 2, 3 and 4 from Section 2.5) are shown in Table 3. Concerning of coverage of Dataset, the Bibit algorithm presents the best result by grouping all the registers and 93.75% of attributes, into, at least, one bicluster. All results have an amount greater than 70% of significant biclusters.

Table 3. Coverage and p-values Measures Results. Source: Search data

Algorithm	Coverage (% of Dataset)	Coverage (% of Registers)	Coverage (% of Attributes)	Biclusters with p-value < 5%
Bibit	93.75	100	93.75	91.96
Bimax ¹	39.58	100	39.58	84.28
Bimax ²	31.96	42.61	75	100
Bimax ³	21.92	31.80	68.75	100
Bimax ⁴	18.85	28.28	66.67	100
Bimax ⁵	17.15	26.56	64.58	100
ISA	37.17	89.2	41.67	73.33
Plaid	33.65	70.22	47.91	71.43

After applying the validation measures, all biclusters containing less than five attributes and significance below 5% were excluded from the analysis (Figure 3(a)). Bibit and BiMax¹ were affected only by the removal of biclusters with low significance. ISA and Plaid were affected by the removal of both significant biclusters and biclusters containing less than 5 attributes. There was no reduction of biclusters in BiMax², BiMax³, BiMax⁴ and BiMax⁵.

The last step, previous to the selection of the best biclusters of each algorithm, consisted in the removal of biclusters containing less than 40 registers and in the classification of the remaining biclusters using previous knowledge. We do not consider relevant groups formed by less than 40 registers. There was only a reduction, from 42.85% to 28.57% of biclusters, in the Plaid, remaining only 2 relevant biclusters. The result of class separation is shown in the Figure 3(b). We observed that only the Plaid algorithm discovered a bicluster containing short-lived.

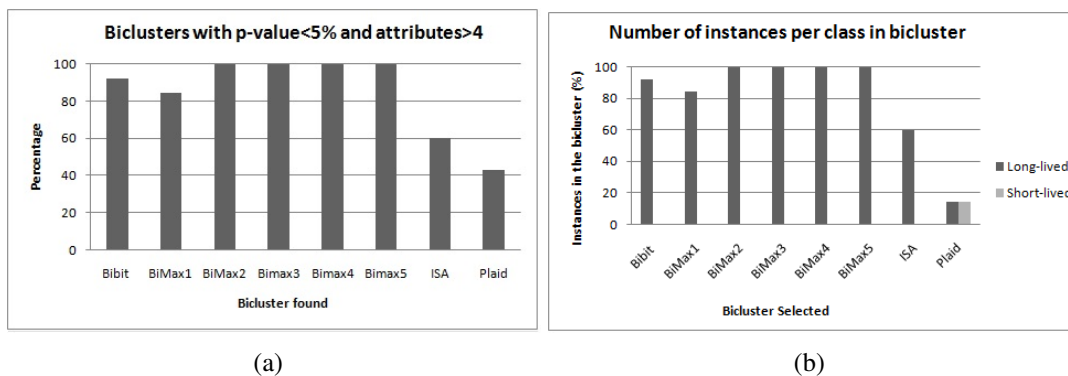


Figure 3. Steps of selection. Source: Search data

The Bibit algorithms and all the executions of the BiMax algorithm had better results when generating an amount greater than 80% of significant biclusters. There was

no reduction of their biclusters during the process of selecting the best results (Figure 3), except by reduction made by remotion of less significant biclusters. It is necessary to emphasize that the BiMax algorithm could not find groups of short-lived during the execution of the algorithm, although subject to several parameters, due to computational limitations. The complete exploration of the algorithm will return optimal short-lived biclusters if they be present in the evaluated dataset.

The criteria for choosing the best long-lived biclusters were based on the largest group of registers containing the least amount of short-lived. The Table 4 shows the best bicluster dimension of each algorithm.

Table 4. Best Biclusters of Long-lived. Source: Search data

Algorithm	# of Registers)	# of Attributes)	% of long-lived
Bibit	197	26	100
Bimax ¹	370	14	85.94
Bimax ²	312	30	98.72
Bimax ³	312	30	98.72
Bimax ⁴	324	30	98.46
Bimax ⁵	324	30	98.46
ISA	339	6	89.38
Plaid	41	5	92.7

We note, through comparison of similarity that, BiMax² and BiMax³ are the same bicluster and, BiMax⁴ and BiMax⁵, are another single bicluster. The Bimax² bicluster is entirely contained in Bimax⁴, except for a single attribute. Approximately 50% of registers of BiMax² and BiMax⁴ belong to BiMax¹.

There is intersection of, approximately, 85.7% of the attributes of the best bicluster discovered by Bibit with the attributes of Bimax¹. There is no intersection of the attributes of the best bicluster discovered by ISA with Bibit's best bicluster, but there is 83.3 % intersection of ISA attributes with the best bicluster of BiMax¹. Concerning to registers grouped by the ISA, there are intersections with biclusters discovered by Bibit and Bimax.

The bicluster classified as short-lived, using prior knowledge and discovered by the Plaid algorithm, has 43 registers and 6 attributes. This bicluster has 60.5% of short-lived registers. This bicluster has an intersection with 5 registers of BiMax¹, and 3 of these registers correspond to long-lived registers grouped as short-lived. Only one register has intersection with Bibit and, similarly, it is a register of the long-lived class grouped as short-lived.

The biclusters containing long-lived and short-lived, discovered by Plaid algorithm, do not intersect any register belonging to the bicluster discovered by ISA algorithm. Through the intercession of the short-lived bicluster with all long-lived biclusters, we found attributes that are present in all biclusters of long-lived and absent bicluster of short-lived. These attributes were considered as those that divide the groups of long-lived and short-lived, considering the input configuration parameters of the Plaid algorithm.

The attributes that separated the classes, subject to parameters configurations of the algorithms in this work, are related to symptoms of depression reported through the questionnaire and limitation caused by health problems.

5. Conclusions

The use of bicluster algorithms applied in the longitudinal study base ELSA demonstrated good separation of groups of long-lived in all the algorithms used, with the majority of the results containing an amount above 70% of long-lived and with assured significance p-value.

Only the Plaid algorithm discovered a significant bicluster containing a class of short-lived. The attributes of this bicluster was used to verify intersection between short-lived bicluster discovered with all long-lived biclusters. As the intersections are minimal, considering the obtained biclusters and the general result, it was possible to define adequately a subset of attributes, which separated the classes of long-lived and short-lived.

In future works, one can execute the Bimax algorithm, in order to explore all the biclusters contained in this dataset, to locate all the optimal biclusters and, possibly, discovery an optimal bicluster of short-lived. A more in-depth study on the way in which the adjustments, defined in the input parameters of the Plaid algorithm, act on the dataset, can lead to the definition of a parameter that allows the discovery of a larger number of short-lived groups. The discovery of groups, generated by the bicluster algorithms, can contribute, in future works, to a later clustering of the registers in biclusters, using the continuous attributes that were not used in this work. This could enable the discovery of groups that share categorical and continuous data.

6. Acknowledgment

The data were made available through the UK Data Archive. ELSA was developed by a team of researchers based at the NatCen Social Research, University College London and the Institute for Fiscal Studies. The data were collected by NatCen Social Research. The funding is provided by the National Institute of Aging in the United States, and a consortium of UK government departments coordinated by the Oce for National Statistics. The developers and funders of ELSA and the Archive do not bear any responsibility for the analyses or interpretations presented here.

This work was conducted during a scholarship supported by the International Cooperation Program CAPES/COFECUB at the PUC-Minas University. Financed by CAPES Brazilian Federal Agency for Support and Evaluation of Graduate Education within the Ministry of education of Brazil.

References

- Ayadi, W., Elloumi, M., and Hao, J.-K. (2009). A biclustering algorithm based on a bicluster enumeration tree: application to dna microarray data. *BioData mining*, 2(1):9.
- Bergmann, S., Ihmels, J., and Barkai, N. (2003). Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical review E*, 67(3):031902.
- Beyer, K. S., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory*, ICDT '99, pages 217–235, London, UK, UK. Springer-Verlag.

- Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103. AAAI Press.
- de França, F. O. (2016). A hash-based co-clustering algorithm for categorical data. *Expert Systems with Applications*, 64:24–35.
- Lazzeroni, L. and Owen, A. (2002). Plaid models for gene expression data. *Statistica sinica*, pages 61–86.
- Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1(1):24–45.
- Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., and Zitzler, E. (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122.
- Ribeiro, C. E. (2017). Mineração de dados longitudinais. um estudo de caso sobre o envelhecimento humano. Master’s thesis, Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte MG.
- Ribeiro, C. E., Brito, L. H. S., Nobre, C. N., Freitas, A. A., and Zárata, L. E. (2017). A revision and analysis of the comprehensiveness of the main longitudinal studies of human aging for data mining research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(3):e1202–n/a. e1202.
- Ribeiro, C. E. and Zárata, L. E. (2017). Data preparation for longitudinal data mining: a case study on human ageing. *Journal of Information and Data Management*, 7(2):116.
- Rodriguez-Baena, D. S., Perez-Pulido, A. J., and Aguilar-Ruiz, J. S. (2011). A bi-clustering algorithm for extracting bit-patterns from binary datasets. *Bioinformatics*, 27(19):2738–2745.
- Saber, H. B. and Elloumi, M. (2014). A comparative study of clustering and biclustering of microarray data. *International Journal of Computer Science & Information Technology*, 6(6):93.
- Saber, H. B. and Elloumi, M. (2015). A novel biclustering algorithm of binary microarray data: Bibincons and bibinalter. *BioData mining*, 8(1):38.
- Veroneze, R., Banerjee, A., and Von Zuben, F. J. (2017). Enumerating all maximal bi-clusters in numerical datasets. *Information Sciences*, 379:288–309.
- Zhao, H., Wee-Chung Liew, A., Z. Wang, D., and Yan, H. (2012). Biclustering analysis for pattern discovery: Current techniques, comparative studies and applications. *Current Bioinformatics*, 7(1).

Planning as Mixed-Horn Formulas Satisfiability

Razer A. N. R. Montaña¹, Bruno C. Ribas²

¹ Professional and Technological Education Sector – Federal University (UFPR)
Curitiba, PR – Brazil.

² Academic Department of Informatics – University of Technology (UTFPR)
Pato Branco, PR – Brazil.

razer@ufpr.br, brunoribas@utfpr.edu.br

Abstract. *Over the course of the last decade, there have been several improvements in the performance of Boolean Satisfiability (SAT) but just a few works focused on Planning as Satisfiability recently. This paper explore a new way to model a planning instance from BLACKBOX and SATPLAN06. This new approach is a mixed-horn formula (MHF) where it can be seen as a conjunction of a Horn formula and a 2-CNF formula. The theoretical complexity to solve a MHF is $O(2^{0.5284n})$. With the theoretical complexity combined with improvements of SAT solvers it is a promising approach to model Planning as Mixed-Horn Formulas Satisfiability. It is observed that this formulation have some improvements with current state-of-the-art SAT solvers.*

1. Introduction

Planning in Artificial Intelligence is an important field of study and an interesting application of Satisfiability to solve generic search problem. The planning problem have the objective to find a sequence of actions from an initial state to reach a given goal. These actions sequences and state sequences attained from actions taken corresponds to assignments of a propositional formula which can be easily generated from the problem description.

In [Kautz and Selman 1992] it is proposed a way to solve planning problems using propositional Satisfiability (SAT), but only after 1996, they demonstrated that modern SAT solvers and better translations provided a competitive approach to planning [Kautz and Selman 1996].

SATPLAN is the planner based on SAT and one of the most successful approach to STRIPS planning. This approach translates a planning problem to SAT problems, one formula to each horizon of actions applied. The first satisfiable formula means that a plan was found.

Efficiency of SAT solver is one of the most important characteristics to success of this logical approach to planning problem. Consequently, encoding the problem is also important, because the information represented can make the solver work better.

SATPLAN06 constructs a planning graph in a GRAPHPLAN [Blum and Furst 1995] fashion, then translates this graph to a set of clauses (CNF) to be solved by a SAT Solver. While the generated formula is UNSAT, the graph

is expanded. When the formula is SAT, the valuation found is translated to the solution of original planning problem.

The classical translation of planning to satisfiability is using axioms generated from planning graph. The choice of axioms leave to different encodings with different characteristics, as presented by [Sideris and Dimopoulos 2010].

While solving time is highly dependent of the SAT solvers, due to better implementation and better solver choices such as: branching heuristics, learning and restart strategies, among others; the way of generating the formula is very important to achieve better solving time, while it is common sense that a minimal formula is better, this is not always the case for current solvers as briefly discussed in [Prestwich 2009]. Also [Hooker 2007] suggests that Constraint Programming and Operational Research, in which SAT exchanges ideas and techniques, are Empirical Science and encoding of problem have to be considered when solving a problem.

In this paper, we show that a planning problem can be translated to a Mixed-Horn Formula (MHF) instance, to be solved by specific SAT solver. MHFs are formulas composed by 2-CNF positive formulas and Horn formulas. It was demonstrated in [Porschen and Speckenmeyer 2007] that SAT in MHF is $\mathcal{O}(2^{0.5284n})$, still NP-Complete, but theoretically better than general CNF SAT.

This paper is organized as follows. In the section 2 we present theoretical foundations and background information of SAT, planning and MHF. In the section 3 we show the proposed translation of a planning problem in MHF SAT, based on available SATPLAN encodings followed by section 4 where we show preliminary results encoding planning problems in MHF. Finally, in section 5 we present conclusion future work.

2. Theoretical Foundations and Background

2.1. Boolean Satisfiability

In Boolean Logic, a formula is said to be in the *Conjunctive Normal Form* (CNF) if it is formed by a sequence of conjunctions, represented by “ \wedge ”, consisting of one or more clauses. Each of these clauses should be a disjunction, represented by “ \vee ”, of one or more literals. A literal is a symbol or the negation of it, used to represent a Boolean statement in logic that can take the value either *True* or *False*. All conjunctions of literals and all disjunctions of literals are in *CNF*, as they can be seen as conjunctions of one-literal clauses and conjunctions of a single clause, respectively. As mentioned, the only propositional connectives in a *CNF* formula are “ \wedge ”, “ \vee ” and “ \neg ”.

Given an assignment to the literals in a CNF formula, the formula is satisfied *iff* all of its clauses are satisfied and a clause is satisfied *iff* at least one of its literals is satisfied. A positive literal is satisfied *iff* the corresponding variable is assigned value *True* and a negative literal is satisfied *iff* the corresponding is assigned to *False*.

The problem of deciding whether a CNF (Boolean) formula is satisfiable, i.e if there exists an assignment to the variables such that the formula is satisfied, is called Boolean *Satisfiability* (SAT). SAT was the first problem that was proved to be NP-complete [Cook 1971].

A detailed description of modern SAT solvers, techniques and CNF encodings can

be found, respectively in [Darwiche and Pipatsrisawat 2009, Marques-Silva et al. 2009, Prestwich 2009].

2.2. Planning

A *planning problem*, in the reasoning about actions area, is defined as the problem to discover a sequence of actions that goes from an initial situation to reach a goal state. According to [Weld 1999], an algorithm that solve these kind of problems must receive as input:

- A description of world entities and actual situation;
- A description of goals;
- A description of allowed actions.

These algorithms return a sequence of actions that reach a goal state from a starting state.

Planning problems were computationally modeled by [Fikes and Nilsson 1971], where they proposed a formal representation that could solve the problem, in conjunction with a search algorithm. This approach is known as STRIPS. Despite simplicity of STRIPS, planning was proved PSPACE-Complete in [Bylander 1994].

The first algorithm that solves planning with satisfactory performance was SATPLAN [Kautz and Selman 1992, Kautz and Selman 1996], that translates the planning problem to satisfiability [Selman et al. 1992] (SAT).

In [Blum and Furst 1995] a new approach is presented, resulting a new planner called *GRAPHPLAN*. This approach represents the planning problem as a planning graph and the resolution is using a backward search. In [Kautz and Selman 1999] SAT-based and graph-based approaches were united, resulting in a smaller instance compared to the generated from *GRAPHPLAN*, it was called BLACKBOX.

In 2004, Kautz and Selman submitted to 2004 International Planning Competition (IPC) a new version of SATPLAN (called SATPLAN04) with some improvements. But in 2006, the SATPLAN06 [Kautz et al. 2006] was implemented with mutex propagation on the planning graph and an encoding with boolean variables for actions and fluents. This allowed harder instances to be solved and avoid some memory problems.

2.3. Planning as Satisfiability

A STRIPS planning problem is a triple $P = \langle I, G, A \rangle$, where I is the set of facts true in initial state, G is the set of facts true in the goal state, an A a set of allowed actions. Each action $a \in A$ has preconditions ($pre(a)$), added effects ($add(a)$) and deleted effects ($del(a)$).

In SAT, propositions are time-stamped to represent actions and facts in different situations. Let $a \in A$ be an action, so $a(t)$ is a decision if the action a is taken or not at time t . The same notion is applied to facts, eg, $f(t)$.

According to [Sideris and Dimopoulos 2010], the direct translation of *GRAPHPLAN* into propositional logic is made using some subset of the following clauses:

1. Unit clauses for initial and goal state;

2. $a(t) \rightarrow f(t)$, for every action a and fact $f \in pre(a)$;
3. $a(t) \rightarrow f(t + 1)$, for every action a and fact $f \in add(a)$;
4. $a(t) \rightarrow \neg f(t + 1)$, for every action a and fact $f \in del(a)$;
5. $f(t) \rightarrow a_1(t - 1) \vee \dots \vee a_m(t - 1)$, for every fact $f \in add(a_i)$ and all actions $a_i, 1 \leq i \leq m$ (including the noops);
6. $\neg f(t) \rightarrow a_1(t - 1) \vee \dots \vee a_m(t - 1) \vee \neg f(t - 1)$, for every fact $f \in del(a_i)$ and all actions $a_i, 1 \leq i \leq m$;
- 7.1 $\neg a_1(t) \vee \neg a_2(t)$, for every pair of action a_1, a_2 , such that the set $del(a_1) \cap pre(a_2)$ is non-empty;
- 7.2 $\neg a_1(t) \vee \neg a_2(t)$, for every pair of action a_1, a_2 , such that the set $del(a_1) \cap add(a_2)$ is non-empty;
- 7.3 $\neg a_1(t) \vee \neg a_2(t)$, if there is a pair of facts $f_1 \in pre(a_1), f_2 \in pre(a_2)$, such that f_1, f_2 are mutually exclusive at time t ;
- 8 $\neg f_1(t) \vee \neg f_2(t)$, for every pair of facts f_1, f_2 that are mutex at time t .

BLACKBOX support different encodings of planning problem, for example:

- BB-7: Clauses 1, 2, 5, 7.1, 7.2, 7.3
- BB-31: Clauses 1, 2, 3, 4, 5, 7.1, 8
- BB-32: Clauses 1, 2, 3, 4, 5, 7.1, 7.2, 7.3, 8

SATPLAN06 also supports different encodings, such as:

- SATPLAN06-3: Clauses 1, 2, 5, 7.1, 7.2, 7.3, 8
- SATPLAN06-4: Clauses 1, 2, 5, 7.1, 7.2, 8

Each encoding have different characteristics and must be chosen carefully to be translated to MHF.

2.4. Mixed-Horn Formulas

Mixed-Horn Formulas (MHF) are formulas composed of a 2-CNF part (clauses containing only two literals) and a Horn part (clauses with at most one positive literal). In our case, the 2-CNF part is composed only by positive literals, called positive monotone 2-CNF. The research in MHF is not new, and there are many problems that can be formulated in terms of MHF, like level-planarity test [Randerath et al. 2001] and graph colorability. Now we are introducing planning as a problem that can be represented directly as MHFs.

It is well known that 2-CNF satisfiability and Horn satisfiability are solvable in linear time [Aspvall et al. 1979, Minoux 1988], but SAT for MHF (MHF-SAT) is NP-Complete as showed in [Porschen and Speckenmeyer 2007]. In [Porschen and Speckenmeyer 2005, Porschen and Speckenmeyer 2007], it was studied satisfiability of Mixed-Horn Formulas, proving that the worst case is $\mathcal{O}(2^{0.5284n})$, where n is the number of variables in the instance.

Porschen and Speckenmeyer also showed there is a polynomial time transformation of an unrestricted CNF to MHF. In our case, this transformation is not necessary, because each horizon of planning problem can be generated directly as MHF, with only one simple flip of some literals and it will be explained in the next section.

Algorithm 1 MHFSAT**Input:** M : MHF Formula**Output:** τ : model foundcompute $P := P(M)$ **if** $P = \emptyset$ **then**return $\tau \leftarrow \text{HornSat}(M)$ **end if**comput graph G_p $\tau \leftarrow \text{nil}; X \leftarrow \text{nil};$ **repeat**compute by $\text{MinVC}(G_p)$ the next minimal vertex cover X of G_p **if** $X \neq \text{nil}$ **then** $\tau \leftarrow \text{HornSat}(M[X])$ **end if****until** $\tau \neq \text{nil}$ OR $X = \text{nil}$ **return** $X \cup \tau$

Algorithm 1 defined by [Porschen and Speckenmeyer 2007] solves MHF-SAT using minimal vertex cover. The function $P(M)$ returns all positive monotone 2-clauses in M , $\text{MinVC}(G)$ generate all minimal vertex covers of a graph G with polynomial delay. The function $\text{HornSat}(H)$ returns a minimal model τ of H if, and only if, H is a satisfiable Horn formula, else returns *nil*.

3. Proposed Translation

This paper proposes a Mixed-Horn formulation to the formulas generated by SATPLAN06 and BLACKBOX hoping we may gain some improvements in solving time with current state-of-the-art SAT solvers. It is important to note that the transformation is cheap to implement, leading no extra overheads to the planner, as shown below.

Analysing the axioms used to generate SAT instance from planning graph, we can generate Mixed-Horn Formulas from, eg, configuration BB-7. In this set of clauses, clauses 1 are unit, clauses 2, 7.1, 7.2 and 7.3 are 2-CNF, and clause 5 can be Horn if we invert interpretation of action taken. If an action a is taken, we represent it as $\neg a$ and if a is not taken we can represent it as a .

The set of axioms (1, 2, 5, 7.1, 7.2 and 7.3), representing actions as described above and converted to clauses, will be:

- 1 Unit clauses for initial and goal state;
- 2 $a(t) \vee f(t)$, for every action a and fact $f \in \text{pre}(a)$;
- 5 $\neg f(t) \vee \neg a_1(t-1) \vee \dots \vee \neg a_m(t-1)$, for every fact $f \in \text{add}(a_i)$ and all actions $a_i, 1 \leq i \leq m$ (including the noops);
- 7.1 $a_1(t) \vee a_2(t)$, for every pair of action a_1, a_2 , such that the set $\text{del}(a_1) \cap \text{pre}(a_2)$ is non-empty;
- 7.2 $a_1(t) \vee a_2(t)$, for every pair of action a_1, a_2 , such that the set $\text{del}(a_1) \cap \text{add}(a_2)$ is non-empty;
- 7.3 $a_1(t) \vee a_2(t)$, if there is a pair of facts $f_1 \in \text{pre}(a_1), f_2 \in \text{pre}(a_2)$, such that f_1, f_2 are mutually exclusive at time t ;

Thus, clauses 1 are unit, clauses 2, 7.1, 7.2 and 7.3 are positive monotone 2-CNF and clauses 5 are Horn. Clearly this set of clauses is MHF (2-CNF and Horn) and can be solved or with general SAT solvers or specific MHF solvers.

To modify SATPLAN06, we used the configuration 4 (clauses 1, 2, 5, 7.1, 7.2 and 8). The clauses converted are the same of BLACKBOX, except for exclusion of clause 7.3 and inclusion of clause 8, that is negated 2-CNF (or negated Horn, depending on interpretation). The clause 8 just have facts, so its generation to MHF is the same:

$$8 \quad \neg f_1(t) \vee \neg f_2(t), \text{ for every pair of facts } f_1, f_2 \text{ that are mutex at time } t.$$

The translation proposed change some literals (all action literals) by its negations. This change makes the formula generated by SATPLAN06 be in MHF format. This formula is equi-satisfiable to the original formula and, when applied to a SAT Solver, have the same behavior, but valuation of action literals must be negated in the subjacent planning problem.

Theorem 1. *Be Φ a theory in CNF. For all literal α in Φ , changing α by $\neg\alpha$ gives a theory Φ' which is equi-satisfiable to Φ .*

Proof. Φ and Φ' are equi-satisfiable if Φ is satisfiable if, and only if, Φ' is satisfiable. If Φ is SAT, then Φ' is SAT too. Let V be a valuation in Φ . If α is *true* (*false*) in V , so, by transformation made, the same valuation with α as *false* (*true*) will be a model of Φ' . If Φ' is SAT, then Φ is SAT too. Let V' be a valuation in Φ' . If α is *true* (*false*) in V' , so, by transformation made, the same valuation with α as *false* (*true*) will be a model of Φ . \square

As showed in [Bylander 1994], planning is PSPACE-Complete. As in CNF, SAT in MHF is NP-Complete and despite of use of MHF representation (where SAT theoretically have better complexity function), solving planning with satisfiability in MHF formulas still remains in PSPACE-Complete class of complexity.

4. Experiments

For evaluation of efficiency of MHF compared to standard CNF encoding we used SATPLAN06 as our planner and used several domains of classical planning (STRIPS) from the ICAPS competition of various years. Our experiments were executed on a Intel Xeon 2.1GHz with 256GB of memory.

Each problem were executed with SATPLAN06 10 times for every combination. For all executions SATPLAN06 was set to run with LINGELING and GLUCOSE SAT solvers, for both encodings: default CNF and MHF. Each formula generated by SATPLAN06 were given a time limit of 3600 seconds to solve and the global time limit (including time spent solving formulas and the planner generating them) was set to 4000 seconds. All execution uses SATPLAN06 with configuration 4, explained in section 2.3.

Tables 1 to 14 shows average time in seconds of 10 executions of each problem. All tables are divided in two different results, one with CNF encoding and other with MHF encoding, in both approaches it is shown average time of SATPLAN06 with LINGELING and GLUCOSE SAT solvers. Inside table, TO means timeout were reached, MEM means memory limit exceeded.

Table 1 Domain: Storage

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	57.688	12.945	56.925	14.300
2	28.897	10.101	35.936	10.155
3	228.196	35.283	227.794	36.409
4	TO	306.438	TO	304.40

Table 2 Domain: Satellite

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	3.046	2.581	3.001	2.560
2	5.367	5.336	5.383	5.332
3	43.690	20.257	53.270	17.865
4	19.904	17.761	30.769	17.657

Table 3 Domain: Blocks

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	.678	.695	.667	.679
2	1.149	1.171	1.128	1.161
3	1.233	1.211	1.216	1.197
4	4.370	4.088	4.323	4.065
5	24.375	29.546	24.321	29.373
6	106.738	161.330	106.054	160.248
7	333.911	494.478	330.293	491.316

Table 4 Domain: Gripper

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	.662	.679	.648	.658
2	1.524	1.453	1.549	1.432
3	3.325	2.461	3.200	2.432
4	10.639	4.833	9.554	4.696
5	36.527	15.645	50.317	16.509
6	239.897	90.239	462.487	90.275
7	2894.689	1236.842	3127.132	2104.082

Table 5 Domain: Depots

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	6.747	6.162	6.816	6.137
2	6.728	6.143	6.799	6.103
3	58.443	82.510	57.055	81.727
4	532.297	494.540	525.645	485.740
5	877.930	820.832	1003.564	813.832

Table 6 Domain: Logistic

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	2.148	1.824	2.116	1.777
2	5.477	4.203	5.418	4.094
3	9.429	7.053	9.400	6.903
4	150.633	44.106	134.581	44.842
5	1398.500	1691.914	543.222	535.163

Table 7 Domain: Drivelog

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	3.913	2.831	3.901	2.820
2	30.701	8.340	25.789	8.343
3	21.816	10.215	23.541	10.114
4	49.446	24.298	50.025	23.838
5	85.416	19.327	98.968	19.924
6	1569.046	972.555	1655.302	978.371

Table 8 Domain: Elevator

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	4.130	3.529	4.375	3.544
2	8.101	4.424	7.690	4.384
3	6.967	4.444	8.822	4.428
4	30.288	10.541	30.886	9.653
5	151.592	39.544	158.124	36.591
6	1177.997	949.610	1125.387	1122.207

Table 9 Domain: TPP

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	31.169	27.620	31.260	27.510
2	55.887	35.541	54.953	35.319
3	76.951	75.865	74.593	76.543
4	1310.279	420.913	1363.944	394.910

Table 10 Domain: Thoughtful Bootstrap

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	110.633	49.679	117.127	50.494
2	35.278	17.435	35.513	17.285
3	55.678	26.905	53.534	26.874
4	24.689	16.111	15.648	15.977
5	131.550	55.693	141.104	53.715

Table 11 Domain: FreeCell

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	1.774	1.768	1.776	1.734
2	35.109	29.692	36.313	29.765
3	9.844	15.917	9.609	15.916
4	402.043	486.056	421.980	367.658
5	2295.060	TO	2151.280	TO

Table 12 Domain: PipesWorld

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	97.892	52.732	97.851	54.316
2	284.710	382.130	288.636	384.086
3	154.948	94.122	163.359	86.735
4	369.755	405.247	348.846	397.876
5	517.795	457.101	481.796	452.125

Table 13 Domain: Childsnack

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	13.692	16.172	13.566	15.994
2	13.691	16.179	13.495	15.991
3	18.711	29.480	18.434	29.238
4	25.335	46.123	24.981	45.756
5	35.531	78.851	35.001	78.547
6	35.483	78.812	34.910	78.360
7	52.280	46.895	51.055	41.637
8	51.976	42.452	50.810	41.675
9	75.915	61.691	74.840	60.234
10	76.071	61.350	74.634	60.196
11	100.243	80.574	98.256	79.517
12	100.081	80.418	98.594	79.988
13	144.287	117.127	142.867	114.805
14	202.641	160.025	193.888	168.339
15	203.591	159.455	192.476	156.243
16	252.118	201.425	251.550	194.885
17	345.853	281.009	345.302	263.331
18	464.029	350.808	457.279	348.500
19	MEM	MEM	MEM	MEM
20	MEM	MEM	MEM	MEM

Table 14 Domain: Rovers

n	CNF		MHF	
	LING	GLUC	LING	GLUC
1	1.097	1.065	1.127	1.084
2	.734	.732	.703	.747
3	1.239	1.198	1.209	1.232
4	.671	.685	.742	.705
5	2.334	1.753	2.099	1.804
6	16.898	4.362	16.734	4.507
7	1.719	1.486	1.714	1.509
8	31.470	4.730	34.578	4.782
9	5.740	4.402	5.758	4.445
10	TO	21.418	TO	20.151
11	10.002	11.325	10.013	11.356
12	12.683	3.357	13.184	3.417
13	TO	162.792	TO	175.692
14	8.779	7.134	8.615	7.109
15	45.830	26.160	38.673	26.424
16	TO	59.306	TO	53.972
17	TO	182.810	TO	169.837
18	TO	88.624	TO	88.513
19	TO	TO	TO	TO
20	TO	TO	TO	TO
21	TO	TO	TO	TO
22	TO	TO	TO	TO
23	TO	TO	TO	TO
24	TO	TO	TO	TO
25	939.990	808.109	938.461	787.637
26	TO	1199.304	TO	1181.282

With these results we can conclude that solving classical planning problems using SAT with MHF formulas is feasible. As we can see, the results using MHF are close to the results using traditional CNF. State-of-the-art SAT solvers are generic solvers for the Satisfiability problem and might not always take advantage of the specific structural information rooted in the formula, therefore a specific SAT solver to solve MHF is needed to enhance solving time, it is possible that small changes to the branching heuristics and learning structures might improve the performance.

Despite the closer times, some problems are solved faster in MHF. In the Logistics domain, table 6, the problem 5 takes 1398.5s to solve with LINGELING in CNF while it takes 543.222s in MHF. The main reason is the last formula (the SAT formula to be solved) that takes 777s to be solved in CNF and is just 67s in MHF. Other domains have this behavior too, but we are still studying this phenomena.

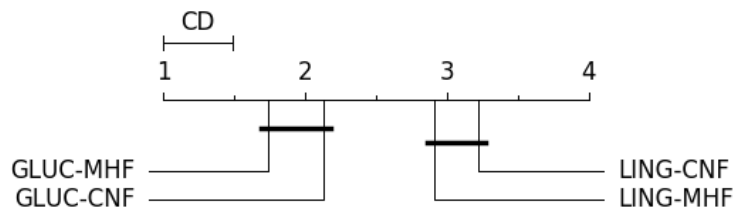


Figure 1. Difference graphic for Nemenyi post-test.

Analysing all executions, we need to know if there is statistic difference between the $K = 4$ configurations and $N = 93$ databases (taking only bases with execution was successfull in all configurations). Friedman test [Demsar 2006] is used to make paired data analysis using performance positions (rank) of all executions. For all N databases, each execution is ranked and a average for each K configurations is calculated.

Although Friedman test use a χ^2 distribution, [Demsar 2006] recommends a less conservative version proposed in [Iman and Davenport 1980], which uses F distribution with $(K - 1)$ and $(K - 1)(N - 1)$ degrees of freedom.

Evidencing the statistic difference, we applied [Nemenyi 1963] post-test to show it. According it, effectiveness of two methods is significantly different if their ranks differs at least for a critical difference value, using a level of significance.

Table 15 shows the average ranking made using all executions.

Table 15 Average ranking Average of execution ranking of formulas. To generate valuable results, we used only formulas solved by both solvers in both encodings. For each database, all executions are ranked (1-4) from best time execution to worst. Then, for each encoding, average of all rankings are calculated.

	CNF		MHF	
	LING	GLUC	LING	GLUC
Average rank	3,21505376344086	2,12903225806452	2,91397849462366	1,74193548387097

Applying Friedman test with $K = 4$ configurations, $N = 93$ databases and significance level at 5%, Friedman statistic results $F_f = 35.59910199$ and $p - value = 1.76238E - 19$, pointing to statistical difference.

To evidence this difference, we applied Nemenyi post-test at critical difference $CD = 0.486370723$, $q_{0.05} = 2.569032073$. Figure 1 shows the resulting difference diagram. In this diagram, x axis represents average rank for each configuration. Lines below x axis connect configurations without statistical difference at 95% of confidence level. Critical difference (CD) is showed above x axis.

From this diagram, we can see there is no statistical difference because of problem encoding, but the solver used.

5. Conclusion

This paper presented a new way to encode Planning as SAT using Mixed-Horn Formulas. It is important to focus in ways to represent problems as techniques and

SAT solver improvements are not the only factor to reduce solving time, as discussed by Hooker [Hooker 2007].

Analysing formulas generated by planners that use satisfiability as engine, we note that clauses are close to MHF. With an equi-satisfiable modification we can generate MHF instead of CNF and this new encoding is now an option to solvers.

MHFs have received much attention recently [Kottler et al. 2008, Porschen et al. 2009]. Many NP-complete problems were proved to have encoding to MHF [Porschen et al. 2009]. In this paper, we show classical planning also can be solved using MHF and have a promising area of study.

In [Porschen and Speckenmeyer 2005, Porschen and Speckenmeyer 2007], they showed that the worst case satisfiability of a MHF is $\mathcal{O}(2^{0.5284n})$ and despite this complexity, it is still NP-complete. They showed too that any SAT instance can be encoded in MHF in polynomial time. In our encoding, the same process that generates CNF is used to generate MHF, without any overhead.

In this paper we show that MHF can be used successfully to encode a planning problem, modified SATPLAN06 to generate MHFs instead normal CNF and executed many experiments using the same domains applied in IPC. The results shows no significant improvements over the traditional CNF encoding, but an implementation of a new SAT solver based in MHF structure can improve our experiments.

We show there is no statistical difference between encodings used, despite some better executions when MHF is applied. This can occurs because the solvers used in MHF encoding are not optimized to take advantage of MHF structure.

As future work, specific SAT solvers for MHF must be used to compare executions. Even a new implementation using specific MHFs for planning can be studied. Besides, as described in [Porschen and Speckenmeyer 2007], we can study the effect of reducing the number of essential variables, and if it is applied to our domain (classical planning).

For now, we are studying the phenomena described in problem Logistics-5, trying to identify domains with this behavior and solving UNSAT formulas faster in MHF.

References

- Aspvall, B., Plass, M. F., and Tarjan, R. E. (1979). A Linear-Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas. *Inf. Process. Lett.*, pages 121–123.
- Biere, A., Heule, M. J. H., van Maaren, H., and Walsh, T., editors (2009). *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.
- Blum, A. and Furst, M. (1995). Fast Planning Through Planning Graph Analysis. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95)*, pages 1636–1642.
- Bylander, T. (1994). The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69(1-2):165–204.

- Cook, S. A. (1971). The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, NY, USA. ACM Press.
- Darwiche, A. and Pipatsrisawat, K. (2009). *Complete Algorithms*, chapter 3, pages 99–130. Volume 185 of [Biere et al. 2009].
- Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.
- Fikes, R. O. and Nilsson, N. J. (1971). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. Technical Report 43r, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025.
- Hooker, J. N. (2007). Good and bad futures for constraint programming (and operations research).
- Iman, R. L. and Davenport, J. M. (1980). Approximations of the critical region of the fbietkan statistic. *Communications in Statistics - Theory and Methods*, 9(6):571–595.
- Kautz, H. and Selman, B. (1996). Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search. In Shrobe, H. and Senator, T., editors, *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 1194–1201, Menlo Park, California. AAAI Press.
- Kautz, H. and Selman, B. (1999). Unifying SAT-based and graph-based planning. In Minker, J., editor, *Workshop on Logic-Based Artificial Intelligence, Washington, DC, June 14–16, 1999*, College Park, Maryland. Computer Science Department, University of Maryland.
- Kautz, H. A. and Selman, B. (1992). Planning as Satisfiability. In *Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI'92)*, pages 359–363.
- Kautz, H. A., Selman, B., and Hoffmann, J. (2006). SatPlan: Planning as satisfiability. In *Abstracts of the 5th International Planning Competition*.
- Kottler, S., Kaufmann, M., and Sinz, C. (2008). A new bound for an NP-hard subclass of 3-SAT using backdoors. In *Proceedings of the 11th international conference on Theory and applications of satisfiability testing, SAT'08*, pages 161–167, Berlin, Heidelberg. Springer-Verlag.
- Marques-Silva, J. P., Lynce, I., and Malik, S. (2009). *Conflict-Driven Clause Learning SAT Solvers*, chapter 4, pages 131–153. Volume 185 of [Biere et al. 2009].
- Minoux, M. (1988). LTUR: a simplified linear-time unit resolution algorithm for horn formulae and computer implementation. *Information Processing Letters*, 29(1):1–12.
- Nemenyi, P. (1963). *Distribution-free Multiple Comparisons*. Princeton University.
- Porschen, S., Schmidt, T., and Speckenmeyer, E. (2009). On Some Aspects of Mixed Horn Formulas. In Kullmann, O., editor, *Theory and Applications of Satisfiability Testing - SAT 2009*, volume 5584 of *Lecture Notes in Computer Science*, pages 86–100. Springer Berlin Heidelberg.

- Porschen, S. and Speckenmeyer, E. (2005). Worst Case Bounds for Some NP-Complete Modified Horn-SAT Problems. In Hoos, H. and Mitchell, D., editors, *Theory and Applications of Satisfiability Testing*, volume 3542 of *Lecture Notes in Computer Science*, pages 251–262. Springer Berlin Heidelberg.
- Porschen, S. and Speckenmeyer, E. (2007). Satisfiability of mixed Horn formulas. *Discrete Appl. Math.*, 155:1408–1419.
- Prestwich, S. (2009). *CNF Encodings*, chapter 2, pages 75–97. Volume 185 of [Biere et al. 2009].
- Randerath, B., Speckenmeyer, E., Boros, E., Hammer, P., Kogan, A., Makino, K., Simone, B., and Cepek, O. (2001). A satisfiability formulation of problems on level graphs. *Electronic Notes in Discrete Mathematics*, 9(0):269 – 277. {LICS} 2001 Workshop on Theory and Applications of Satisfiability Testing (SAT 2001).
- Selman, B., Levesque, H. J., and Mitchell, D. (1992). A New Method for Solving Hard Satisfiability Problems. In Rosenbloom, P. and Szolovits, P., editors, *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 440–446, Menlo Park, California. AAAI Press.
- Sideris, A. and Dimopoulos, Y. (2010). Constraint Propagation in Propositional Planning. In Brafman, R. I., Geffner, H., Hoffmann, J., and Kautz, H. A., editors, *ICAPS*, pages 153–160. AAAI.
- Weld, D. S. (1999). Recent advances in AI planning. *AI Magazine*, 20(2):93–123.

Uma Hiper-Heurística para o Problema Quadrático de Alocação Multiobjetivo

Geovani Ferreira Antunes¹, Richard Aderbal Gonçalves¹, Sandra Mara Venske¹, Carolina Paula de Almeida¹

¹Departamento de Ciência da Computação – Universidade Estadual do Centro-Oeste (UNICENTRO)

Rua Simeão Camargo Varela de Sá, 03 - Vila Carli – 85040-080
Guarapuava – PR – Brasil

geovanyantunes@hotmail.com

Abstract. *Most optimization problems present more than one objective function to be minimized or maximized. Such problems are part of the class of Multiobjective Optimization Problems (MOPs). The Multi-objective Quadratic Assignment Problem (mQAP) is a complex optimization problem with many applications in the real world, such as in hospital layouts. From the variety of techniques that can be applied in the POM's solution emerges the difficulty of determining which algorithm or which operators are the most efficient for a given problem. In this context, hyper-heuristics have been applied to, automatically, select or generate heuristics for the solution of complex problems. This work proposes the implementation of a Hyper-Heuristic (HH) for the selection of operators in the MOEA/D algorithm for the treatment of mQAP. The results obtained considering instances of the Knowles & Corne benchmark attest to the efficiency of the selection performed by the HH.*

Resumo. *Grande parte dos problemas de otimização apresentam mais de uma função objetivo a ser minimizada ou maximizada. Tais problemas fazem parte da classe de Problemas de Otimização Multiobjetivo (POMs). O Problema Quadrático de Alocação Multiobjetivo (mPQA) é um problema de otimização complexo com muitas aplicações no mundo real, como em layouts hospitalares. Da variedade de técnicas que podem ser aplicadas na solução de POMs, emerge a dificuldade de determinar qual algoritmo ou quais operadores são os mais eficientes para um determinado problema. Neste contexto as Hiper-Heurísticas (HHs) têm sido aplicadas para, automaticamente, selecionar ou gerar heurísticas para a solução de problemas complexos. Este trabalho propõe a implementação de uma HH para a seleção de operadores no algoritmo MOEA/D para o tratamento do mPQA. Os resultados obtidos considerando instâncias do benchmark de Knowles & Corne atestam a eficiência da seleção realizada pela HH.*

1. Introdução

O PQA (Problema Quadrático de Alocação) considera um conjunto de n instalações e um conjunto de n localidades. Para cada par de localidades é especificada uma distância enquanto para cada par de instalações é especificado um valor de fluxo. O

fluxo representa a quantidade de suprimentos transportados entre as duas instalações. O problema consiste em associar todas as instalações a diferentes localidades com o objetivo de minimizar a soma das distâncias multiplicadas pelo fluxo correspondente [Koopmans and Beckmann 1957, Sahni and Gonzales 1976]. O mPQA (Problema Quadrático de Alocação Multiobjetivo) estende o caso mono-objetivo para situações nas quais dois ou mais tipos de fluxos estejam presentes, por exemplo no problema do layout de um hospital pode-se minimizar simultaneamente o fluxo/distância entre médicos, pacientes, funcionários, visitas e equipamentos farmacêuticos [Knowles and Corne 2002]. Para isto, o problema considera múltiplas matrizes de fluxos.

Na ausência de métodos exatos, métodos alternativos tais como as técnicas de inteligência computacional, têm se tornado populares na solução destes problemas práticos de engenharia [Coello et al. 2007]. Atualmente existem diversas heurísticas e metaheurísticas disponíveis para o tratamento de problemas de otimização multiobjetivo [Zhou et al. 2011], porém a maior parte delas requer grandes conhecimentos para serem utilizadas de maneira eficiente. As Hiper-Heurísticas surgiram no contexto de otimização exatamente para diminuir a quantidade de conhecimento necessário para a solução de problemas complexos de otimização. Uma Hiper-Heurística é uma metodologia de alto nível para a seleção ou geração automática de heurísticas para a resolução de problemas complexos [Burke et al. 2013]. Uma diferença importante entre as Hiper-Heurísticas e as metaheurísticas é o espaço de busca onde elas operam: as Hiper-Heurísticas fazem uma busca no espaço das heurísticas enquanto as metaheurísticas fazem uma busca no espaço das soluções [Chakhlevitch and Cowling 2008].

Neste trabalho é implementada uma Hiper-Heurística baseada em *Multi-Armed Bandit* (MAB) para a seleção de operadores de cruzamento. Ela é aplicada na solução do problema quadrático de alocação multiobjetivo. O principal objetivo deste trabalho é demonstrar que uma Hiper-Heurística é capaz de atingir resultados comparáveis e até superiores àqueles obtidos por meio da seleção de operadores feita manualmente para o problema tratado.

O restante das seções deste trabalho estão organizadas conforme a seguir. Na Seção 2 é apresentada a definição formal de um POM e o algoritmo MOEA/D que é utilizado como base para a construção da HH. O problema abordado, o mPQA é descrito na seção 3. Uma breve descrição das Hiper-Heurísticas pode ser encontrada na seção 4. A técnica proposta é descrita na Seção 5 enquanto a Seção 6 apresenta os experimentos e os resultados obtidos com os algoritmos implementados, comparando-os com algoritmos da literatura. Finalmente, a Seção 7 contém as conclusões e trabalhos futuros.

2. Otimização Multiobjetivo e MOEA/D

Um Problema de Otimização Multiobjetivo (POM) pode ser definido como: Minimizar $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x}))$ sujeito a $g_i(\mathbf{x}) \leq 0$, $i = \{1, \dots, G\}$, $h_j(\mathbf{x}) = 0$, $j = \{1, \dots, H\}$ e $\mathbf{x} \in \Omega$, tal que o inteiro $M \geq 2$ é o número de objetivos. Uma solução para o problema minimiza (ou maximiza) os componentes de $f(\mathbf{x})$ tal que \mathbf{x} é um vetor de variáveis de decisão n -dimensional $\mathbf{x} = (x_1, \dots, x_n) \in \Omega$.

O MOEA/D é um algoritmo evolucionário multiobjetivo que aplica uma abordagem de decomposição para transformar um POM em vários subproblemas de otimização com um único objetivo. O objetivo de cada subproblema é uma agregação ponderada

de todos os objetivos individuais do POM. Cada subproblema é resolvido simultaneamente levando em conta as relações de vizinhança entre os subproblemas. A vizinhança é definida em termos das distâncias entre os vetores de peso de agregação. O algoritmo proposto neste trabalho utiliza algoritmo MOEA/D com alocação dinâmica de recursos (MOEA/D-DRA) [Zhang et al. 2009] para alocar diferentes esforços computacionais para cada subproblema. Um valor de utilidade é calculado para a solução de cada subproblema e o esforço computacional alocado aos agentes é baseado nesses valores. Além disso, este trabalho combina a estrutura MOEA/D-DRA com as Hiper-Heurísticas para resolver o problema quadrático de alocação multiobjetivo.

3. Problema Quadrático de Alocação

O PQA (Problema Quadrático de Alocação) do inglês *Quadratic Assignment Problem* foi proposto em 1957 por Koopmans e Beckmann, com aplicação na economia, e pode ser aplicado em muitos problemas do mundo real, como na pesquisa operacional, análise estatística, problemas de otimização combinatória como o do caixeiro viajante, particionamento, isomorfismo de grafos, minimizar o número de componentes em um circuito, na alocação de instalações, logística e em projetos de hospitais [Loiola et al. 2007].

Esse é um dos problemas mais difíceis encontrados na literatura em otimização combinatória, visto que sua complexidade envolve trocas fatoriais, sendo classificado como NP-difícil [Sahni and Gonzales 1976]. Para instâncias com tamanho a partir de 30 localidades, resolver em tempo viável não é possível devido ao alto custo computacional, desta maneira justifica-se a utilização de métodos de aproximação para resolvê-lo.

O problema consiste em alocar instalações a localidades de maneira a minimizar os custos levando em consideração a distância, entre as localidades e o fluxo entre as instalações, quanto maior for o fluxo entre as instalações mais importante é alocá-las em localidades próximas. Assim minimizando a soma das distâncias multiplicadas pelo fluxo.

3.1. Problema Quadrático de Alocação Multiobjetivo - mPQA

O mPQA (*Multi-objective Quadratic Assignment Problem*) é a versão multiobjetivo do PQA que foi introduzida em 2002 por Knowles e Corne na qual diferentes tipos de fluxo são considerados, tendo assim várias matrizes de fluxo ou também várias matrizes de distâncias sendo consideradas. Considerando para as localidades uma matriz quadrada de ordem n $A = \{a_{ij}\}$ e m para os fluxos matrizes quadradas de ordem n $B_k = \{b_{rs}^k\}$, $k = 1, \dots, m$, o mPQA pode ser formulado como [Knowles and Corne 2002]:

$$\begin{aligned} \text{Minimizar } \mathbf{C}(\pi) &= \{C^1(\pi), C^2(\pi), \dots, C^m(\pi)\} \\ \pi &\in \mathbf{P}(n) \\ \text{tal que } \mathbf{C}^k(\pi) &= \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{\pi_i \pi_j}^k, \quad 1 \leq k \leq T \end{aligned} \quad (1)$$

tal que a_{ij} é a distância entre as localidades i e j , b_{ij}^k é o k -ésimo fluxo entre as instalações i e j e π_i representa a posição da instalação i na permutação $\pi \in \mathbf{P}(n)$, onde $\mathbf{P}(n)$ é o conjunto de todas as permutações de $\{1, 2, \dots, n\}$, C é o custo de π e T é o número de objetivos.

Algumas abordagens para o tratamento do mPQA são: em 2014 [Drugan 2014] apresenta um algoritmo para gerar instâncias para o mPQA com uma solução ótima conhecida e mostra, usando experimentos numéricos, que é mais difícil solucionar as instâncias geradas do que instâncias aleatórias. Algoritmos híbridos combinando algoritmos transgenéticos e otimização multiobjetivo evolucionária para tratar o mPQA foram utilizados em [de Almeida et al. 2014]. O trabalho mostra a superioridade de Algoritmos Evolucionários Multiobjetivo (AEMOs) baseados em decomposição com relação aos baseados em dominância de Pareto. Em [Ceberio et al. 2015], instâncias do PQA foram transformadas em instâncias do mPQA e resolvidas com o algoritmo NSGA-II. O trabalho mostrou que algumas instâncias podem ser melhor resolvidas com o aumento do número de objetivos.

4. Hiper-Heurísticas

Hiper-Heurísticas são heurísticas de alto nível com o propósito de automatizar o processo de seleção ou geração de heurísticas (denominadas heurísticas de baixo nível) para resolver de forma eficiente problemas de busca [Bilgin et al. 2006]. As Hiper-Heurísticas podem ser classificadas como sendo de seleção ou geração. As Hiper-Heurísticas de seleção visam escolher a melhor heurística a ser aplicada em cada momento do processo de busca enquanto as Hiper-Heurísticas de geração visam criar novas heurísticas para resolver o problema em questão. Devido aos bons resultados apresentados este trabalho tem foco nas Hiper-Heurísticas de seleção [Burke et al. 2013].

As decisões tomadas pelas Hiper-Heurísticas acontecem de acordo com os resultados obtidos durante o processo de busca ou antes da busca iniciar e, de acordo com este critério, são classificadas em *on-line* ou *off-line*, respectivamente. A vantagem do método *on-line* é permitir que a Hiper-Heurística se adapte dinamicamente às condições correntes do processo de otimização. Portanto, neste projeto, optou-se pelo desenvolvimento de uma Hiper-Heurística com aprendizado *on-line*. As Hiper-Heurística também devem decidir quando uma nova solução gerada deve ou não ser aceita. Esse componente das Hiper-Heurísticas é denominado política ou heurística de aceitação [Bilgin et al. 2006].

4.1. Multi-Armed Bandit

Considere uma máquina com um determinado número de alavancas que ao serem puxadas retornam uma recompensa para o apostador. O padrão (distribuição) das recompensas não é conhecido *a priori* e pode variar em função do tempo. O objetivo do apostador é maximizar seus lucros ao longo de várias tentativas. Para resolver esse problema, conhecido como MAB (*Multi-Armed Bandit*), é importante balancear a intensificação (seleção da alavanca com o melhor retorno empírico) com a exploração (escolha de diferentes alavancas para obter melhores informações sobre as mesmas) [Robbins 1985]. Vários problemas de otimização e aprendizado do mundo real podem ser modelados utilizando essa abordagem, em que se deseja maximizar ganhos ou minimizar perdas, como em testes clínicos por exemplo, onde se procura experimentar tratamentos diferentes, minimizando as consequências negativas aos pacientes ou maximizando o uso de tratamentos mais eficientes [Hardwick et al. 1991].

Cada ação, ativação de uma alavanca, retorna informações sobre a ação tomada, geralmente na forma do lucro obtido com a ação. Essas informações podem ser utilizadas para guiar as decisões futuras com relação a qual ação tomar, visto que quanto

mais ações forem executadas mais informações sobre o problema são acumuladas. O problema está em balancear a maximização dos ganhos baseado no conhecimento que já foi adquirido e explorar novas alternativas para melhorar o conhecimento sobre o problema (intensificação *versus* exploração). Assim determinar a melhor estratégia de seleção das alavancas é o objetivo do MAB, ou seja, determinar a cada nova ação qual alavanca deve ser ativada [Vermorel and Mohri 2005].

Se tratando de HHs pode-se considerar que cada heurística de baixo nível (operador) é uma alavanca cujo retorno é inicialmente desconhecido, ou seja, não se tem a informação se a aplicação deste operador trará ou não, melhorias na solução. As HHs tem como objetivo escolher a melhor heurística para ser aplicada além de explorar o uso das demais heurísticas afim de obter informações sobre a qualidade destas.

Para resolver o MAB existem diversos algoritmos, neste trabalho foi utilizado um algoritmo do tipo *Upper Confidence Bound* (UCB) [Auer 2003], particularmente o UCB1, o primeiro dos algoritmos UCBs propostos. O algoritmo estima com alta probabilidade o limite superior para a recompensa de cada alavanca e escolhe a alavanca com maior limite superior. A exploração acontece para aumentar a confiabilidade dos limites estimados. O UCB1 foi escolhido devido às suas propriedades teóricas: ele é assintoticamente ótimo.

5. Algoritmo Proposto - Uma HH para o mPQA

Esta seção descreve a implementação da Hiper-Heurística de seleção baseada em *Multi-Armed Bandit* para o Problema Quadrático de Alocação Multiobjetivo. O Algoritmo 1 apresenta o pseudocódigo da HH implementada, nomeada MOEA/D-DRA_{MAB}, utilizando como base o algoritmo MOEA/D-DRA [Li and Zhang 2009]. O uso de MAB juntamente com o algoritmo MOEA/D já foi empregado na solução de POMs contínuos [Gonçalves et al. 2015, Li et al. 2014]. Portanto, um diferencial desta proposta reside na natureza do problema tratado.

Os primeiros passos do algoritmo correspondem à inicialização das estruturas utilizadas. Os vetores de pesos, são gerados uniformemente conforme a técnica descrita em [Zhang et al. 2009]. Os vetores de pesos mais próximos de cada vetor λ^i são incluídos na vizinhança de λ^i (B(i)). A população inicial de cromossomos é gerada de maneira aleatória e o *i*-ésimo cromossomo (x^i) é associado ao *i*-ésimo vetor de pesos. O ponto ideal empírico (z) é iniciado com o valor mínimo para cada objetivo encontrado na população inicial. O valor ponto empírico *nadir* é iniciado com o valor máximo para cada objetivo encontrado na população inicial.

Após as inicializações, o algoritmo entra em seu laço principal (Passos de 6 a 33). O primeiro passo do laço seleciona N_e para cada cromossomo. N_e corresponde a um conjunto de índices de cromossomos usados durante os passos de reprodução e atualização. Ele é composto por índices dos cromossomos da vizinhança (B(i)) - com probabilidade d - ou por índices de cromossomos de toda a população - com probabilidade $1,0 - d$. Os indivíduos aos quais os operadores genéticos serão aplicados são selecionados aleatoriamente de N_e .

No passo 12 a escolha do operador mais adequado é feita pela HH e no passo 13 uma nova solução é gerada utilizando este operador. A mutação *Swap* é aplicada na nova solução no passo 14. Os dois laços seguintes (passos 14 a 21) atualizam os pontos

Algoritmo 1 Pseudocódigo do MOEA/D-DRA_{MAB}

```

1: Gerar os vetores de pesos  $\lambda^i = (\lambda_1^i, \lambda_2^i, \dots, \lambda_T^i)$ ,  $i = 1, \dots, Sp$ 
2: Calcular a distância Euclidiana entre os pares de vetores de pesos e encontre os
   C vetores mais próximos  $(\lambda^i, \dots, \lambda^{i_C})$  para cada vetor de pesos  $\lambda^i$ . Para  $i =$ 
    $1, \dots, Sp$ ,  $\mathbf{B}(i) = \{i_1, \dots, i_C\}$ 
3: Gerar a população inicial  $\mathbf{x}^1, \dots, \mathbf{x}^{Sp}$ , aleatoriamente.  $\mathbf{VF}^i = \mathbf{F}(\mathbf{x}^i)$ . Cada  $\mathbf{x}^i$  e  $\mathbf{VF}^i$  é
   associado a um vetor de pesos  $\lambda^i$ 
4: Inicializar  $\mathbf{z}$ :  $z_j = \min_{\text{objetivo}_j}(\mathbf{VF}^i)$ ,  $i = 1, \dots, Sp$ ,  $j = 1, \dots, T$ 
5: Inicializar nadir:  $nadir_j = \max_{\text{objetivo}_j}(\mathbf{VF}^i)$ ,  $i = 1, \dots, Sp$ ,  $j = 1, \dots, T$ 
6: repeat
7:   for cada  $x^i$  do
8:     Gerar rand em U[0,1]
9:     if rand <  $\delta$  then  $Ne = \mathbf{B}(i)$ 
10:    else  $Ne = \{1, \dots, Sp\}$ 
11:    end if
12:    Selecionar o Operador utilizando a HH
13:    Gerar nova solução  $\mathbf{x}'$  por meio do operador selecionado
14:    Aplicar mutação Swap na solução  $\mathbf{x}'$ 
15:    for cada  $j = 1, \dots, T$  do
16:      if  $z_j > F_j(\mathbf{x}'')$  then  $z_j = F_j(\mathbf{x}'')$ 
17:      end if
18:    end for
19:    for cada  $j = 1, \dots, T$  do
20:      if  $nadir_j < F_j(\mathbf{x}'')$  then  $nadir_j = F_j(\mathbf{x}'')$ 
21:      end if
22:    end for
23:    contador = 0
24:    while contador < maxAtualizações e  $\exists$  índices não selecionados de  $Ne$  do
25:      selecionar aleatoriamente um índice  $k$  de  $Ne$ 
26:      if  $g^{te}(\mathbf{x}'' | \lambda^k, \mathbf{z}, \mathbf{nadir}) \leq g^{te}(\mathbf{x}^k | \lambda^k, \mathbf{z}, \mathbf{nadir})$  then
27:         $\mathbf{x}^k = \mathbf{x}''$ 
28:         $\mathbf{VF}^k = \mathbf{F}(\mathbf{x}'')$ 
29:        contador = contador + 1
30:      end if
31:    end while
32:  end for
33: until Critério de parada satisfeito

```

especiais z e *nadir*. Se o novo cromossomo tem um valor para alguma função objetivo que é melhor do que aquele armazenado no ponto ideal, então o ponto z é atualizado com este valor. Analogamente, se o cromossomo tem um valor para alguma função objetivo que seja pior que o valor armazenado no ponto *nadir* então *nadir* é atualizado com este valor.

O último laço (passos 23 a 30) atualiza a população (seleção). De acordo com a escolha feita nos passos de 8 a 11 (Ne), a vizinhança ou a população como um todo é

atualizada. Para evitar a proliferação de x' na maior parte da população é utilizado um número máximo de atualizações (maxAtualizações). A atualização acontece conforme a seguir: enquanto o número de atualizações é menor que o número máximo de atualizações e existem índices de Ne não selecionados, um índice é escolhido aleatoriamente de Ne . Se x' tem um melhor valor para a função de Tchebycheff [Li and Zhang 2009] do que x^k (ambos fazendo uso do k -ésimo vetor de pesos - λ^k) então ele substitui x^k e o número de cromossomos atualizados é incrementado. Finalmente, o critério de parada é testado o qual corresponde a um número máximo de avaliações da função objetivo.

5.1. Descrição da Hiper-Heurística Implementada

Os principais componentes de uma hiper-heurística, conforme a Seção 4 são: mecanismo de seleção, critério de aceitação, critério de parada e as heurísticas de baixo nível, implementadas neste trabalho como operadores. Uma descrição da implementação de cada um destes componentes é apresentada abaixo.

5.1.1. Heurística de Seleção

Para a implementação do mecanismo de seleção foi considerado um método UCB [Auer et al. 2002]. A métrica utilizada para medir a qualidade dos operadores foi a *Fitness Improvement Rate* (FIR), sendo definida pela Equação 2 [Gonçalves et al. 2015]:

$$FIR_{op,t} = \frac{g^{te}(x^i|\lambda^i, z^*) - g^{te}(y|\lambda^i, z^*)}{g^{te}(x^i|\lambda^i, z^*)} \quad (2)$$

tal que $FIR_{op,t}$ é a recompensa do operador op no tempo t , g^{te} o *fitness* utilizando Tchebycheff, com x^i sendo o pai, y o filho, λ o vetor de pesos de i e z^* o ponto de referência. Para resolver o problema da variação de melhoria no *fitness* durante o processo de busca, uma janela deslizante é utilizada para guardar os operadores aplicados recentemente. A fim de aplicar o crédito ao devido operador, uma soma de todos os valores de FIR contidos na janela deslizante é efetuada, então um fator de decaimento é aplicado para que os melhores operadores tenham maiores chances, por fim esse valor é normalizado, gerando o *Fitness-Rate-Ranking* (FRR). O FRR é então utilizado pela seleção de operadores, por intermédio do UCB1. O operador escolhido é aquele que maximiza a Equação 3 [Auer et al. 2002].

$$FRR + C \times \sqrt{\frac{2 \times \ln \sum_{j=1}^K n_{j,t}}{n_{i,t}}} \quad (3)$$

tal que FRR é uma estimativa de qualidade empírica (intensificação), i é o identificador do i -ésimo operador, t é o tempo, C um fator para controlar a intensificação e a exploração do processo de aprendizagem, os termos na raiz quadrada representam as incertezas associadas a cada operador. O número de vezes que o operador i foi utilizado é denotado por n e K o conjunto de operadores.

5.1.2. Critérios de Aceitação e de Parada

Os critérios de aceitação e de parada foram implementados conforme descrito no Algoritmo 1. Uma solução é aceita caso seja selecionada para compor a população da próxima geração do MOEA/D-DRA (passos de 24 a 31 do Algoritmo 1). O critério de parada é estabelecido pelo número máximo de avaliações da função objetivo.

5.1.3. Heurísticas de Baixo Nível - Operadores Genéticos

Considerando o mPQA, é importante a utilização de heurísticas adequadas para esta classe de problemas. Desta forma foram utilizadas as seguintes heurísticas de *Crossover* [Larrañaga et al. 1999]: *Cycle Crossover (CX)*, *Order Crossover (OX)*, *Partially-Mapped Crossover (PMX)*, e a heurística de mutação *Swap*. O CX identifica ciclos que estão contidos nos pais e os repassa para os filhos, possui a característica de gerar filhos mais distintos e assim, diversificando a população. O OX mantém a ordem relativa dos valores, um segmento dos pais é repassado aos filhos e a ordem relativa do outro pai procura ser preservada, sendo importante quando a posição relativa é mais importante que a posição absoluta. Enquanto que no PMX a posição é mais importante que a ordem relativa, essa heurística procura guardar a ordem e a posição dos pais e reproduz essa ordem nos filhos, o restante é replicado e caso haja repetição a posição mais próxima é selecionada [Umbarkar and Sheth 2015]. A mutação *Swap* por sua vez, seleciona duas posições aleatoriamente e realiza a troca de posição entre elas, gerando um novo indivíduo. Após a aplicação do operador selecionado pela HH, a mutação *Swap* é realizada.

6. Simulações e Resultados

Esta seção apresenta os resultados para o conjunto de instâncias de Knowles & Corne [Knowles and Corne 2003]. As simulações são realizadas com as 22 instâncias disponíveis em <http://dbkgroup.org/knowles/mQAP/> e correspondem a problemas com 10, 20 e 30 instalações com 2 e 3 matrizes de fluxo. Os resultados representam 30 execuções independentes dos algoritmos. A Subseção 6.1 apresenta o ajuste de parâmetros. A Subseção 6.2 compara a qualidade dos resultados da Hiper-Heurística proposta (MOEA/D-DRA_{MAB}) com relação aos operadores aplicados isoladamente.

6.1. Ajuste dos Parâmetros

As Tabelas 1, 2 e 3 contêm a média e o desvio padrão para o indicador Hipervolume para o ajuste dos seguintes parâmetros: tamanho da população, taxa de mutação e taxa de cruzamento. Nas referidas tabelas os valores em cinza escuro significam os melhores resultados e os valores em cinza claro correspondem ao segundo melhor resultado. A Tabela 1 exibe os resultados do indicador Hipervolume para diferentes tamanhos de população. A população com 600 soluções obteve os melhores resultados para onze instâncias. Desta forma, a população foi fixada com 600 soluções para os testes subsequentes.

A Tabela 2 apresenta os resultados obtidos pelo indicador Hipervolume para diferentes taxas de mutação. A taxa de 100% obteve os melhores resultados para 19 instâncias e esta taxa foi aplicada nos demais testes.

Tabela 1. Tamanho da População - Média e desvio padrão para o Hipervolume.

	100 soluções	300 soluções	600 soluções
KC10-2fl-1uni.dat	7.82e - 01 _{3.5e-02}	7.86e - 01 _{2.7e-02}	7.95e - 01 _{2.8e-02}
KC10-2fl-1rl.dat	7.94e - 01 _{4.9e-03}	7.91e - 01 _{1.4e-02}	7.95e - 01 _{4.1e-03}
KC10-2fl-2uni.dat	3.12e - 01 _{2.3e-01}	3.87e - 01 _{2.5e-01}	5.37e - 01 _{2.8e-01}
KC10-2fl-2rl.dat	8.61e - 01 _{5.1e-02}	8.81e - 01 _{2.8e-02}	8.87e - 01 _{2.2e-02}
KC10-2fl-3uni.dat	6.12e - 01 _{4.5e-03}	6.12e - 01 _{4.5e-03}	6.12e - 01 _{4.9e-03}
KC10-2fl-3rl.dat	7.96e - 01 _{2.2e-02}	8.04e - 01 _{1.8e-02}	8.09e - 01 _{1.4e-02}
KC10-2fl-4rl.dat	9.03e - 01 _{3.2e-02}	9.12e - 01 _{5.0e-03}	9.12e - 01 _{5.0e-03}
KC10-2fl-5rl.dat	9.15e - 01 _{2.3e-02}	9.22e - 01 _{1.3e-02}	9.27e - 01 _{1.1e-02}
KC20-2fl-1uni.dat	6.68e - 01 _{2.8e-02}	6.64e - 01 _{2.4e-02}	6.59e - 01 _{2.8e-02}
KC20-2fl-1rl.dat	6.95e - 01 _{3.5e-02}	7.05e - 01 _{2.1e-02}	6.99e - 01 _{3.1e-02}
KC20-2fl-2uni.dat	4.94e - 01 _{1.3e-01}	5.04e - 01 _{1.7e-01}	5.17e - 01 _{1.6e-01}
KC20-2fl-2rl.dat	7.50e - 01 _{4.2e-02}	7.70e - 01 _{3.3e-02}	7.64e - 01 _{3.9e-02}
KC20-2fl-3uni.dat	6.10e - 01 _{8.8e-03}	6.07e - 01 _{7.1e-03}	5.99e - 01 _{7.6e-03}
KC20-2fl-3rl.dat	7.77e - 01 _{2.1e-02}	7.89e - 01 _{2.0e-02}	7.94e - 01 _{8.8e-03}
KC20-2fl-4rl.dat	8.14e - 01 _{3.7e-02}	8.13e - 01 _{4.2e-02}	8.20e - 01 _{3.4e-02}
KC20-2fl-5rl.dat	7.68e - 01 _{4.5e-02}	7.91e - 01 _{2.9e-02}	7.94e - 01 _{3.7e-02}
KC30-3fl-1uni.dat	5.06e - 01 _{3.9e-02}	5.08e - 01 _{1.7e-02}	4.95e - 01 _{2.6e-02}
KC30-3fl-1rl.dat	6.31e - 01 _{3.5e-02}	6.61e - 01 _{3.0e-02}	6.43e - 01 _{3.7e-02}
KC30-3fl-2uni.dat	4.65e - 01 _{7.1e-02}	4.44e - 01 _{6.4e-02}	4.45e - 01 _{5.8e-02}
KC30-3fl-2rl.dat	6.28e - 01 _{4.4e-02}	6.22e - 01 _{3.9e-02}	6.28e - 01 _{3.8e-02}
KC30-3fl-3uni.dat	5.39e - 01 _{1.7e-02}	5.60e - 01 _{1.8e-02}	5.50e - 01 _{1.9e-02}
KC30-3fl-3rl.dat	5.96e - 01 _{2.4e-02}	6.26e - 01 _{1.9e-02}	6.18e - 01 _{1.9e-02}

Tabela 2. Mutação - Média e desvio padrão para o Hipervolume.

	50%	75%	100%
KC10-2fl-1uni.dat	8.32e - 01 _{1.9e-02}	8.41e - 01 _{1.4e-02}	8.65e - 01 _{8.3e-03}
KC10-2fl-1rl.dat	7.81e - 01 _{4.8e-03}	7.83e - 01 _{3.1e-03}	7.85e - 01 _{2.5e-03}
KC10-2fl-2uni.dat	5.66e - 01 _{3.1e-01}	7.54e - 01 _{2.9e-01}	9.85e - 01 _{8.0e-02}
KC10-2fl-2rl.dat	9.06e - 01 _{1.5e-02}	9.20e - 01 _{8.8e-03}	9.28e - 01 _{5.1e-03}
KC10-2fl-3uni.dat	6.33e - 01 _{3.6e-03}	6.34e - 01 _{3.2e-03}	6.37e - 01 _{1.9e-03}
KC10-2fl-3rl.dat	8.16e - 01 _{1.1e-02}	8.21e - 01 _{8.5e-03}	8.28e - 01 _{5.5e-03}
KC10-2fl-4rl.dat	7.37e - 01 _{2.7e-03}	7.37e - 01 _{2.4e-03}	7.38e - 01 _{1.5e-03}
KC10-2fl-5rl.dat	9.33e - 01 _{1.1e-02}	9.36e - 01 _{5.7e-03}	9.38e - 01 _{2.7e-03}
KC20-2fl-1uni.dat	7.05e - 01 _{2.1e-02}	7.12e - 01 _{2.1e-02}	7.45e - 01 _{1.4e-02}
KC20-2fl-1rl.dat	7.64e - 01 _{1.8e-02}	7.90e - 01 _{2.7e-02}	8.13e - 01 _{2.3e-02}
KC20-2fl-2uni.dat	4.19e - 01 _{1.1e-01}	4.57e - 01 _{1.3e-01}	6.50e - 01 _{8.5e-02}
KC20-2fl-2rl.dat	8.18e - 01 _{3.0e-02}	8.18e - 01 _{2.5e-02}	8.34e - 01 _{2.1e-02}
KC20-2fl-3uni.dat	5.95e - 01 _{6.8e-03}	6.00e - 01 _{7.0e-03}	6.04e - 01 _{5.0e-03}
KC20-2fl-3rl.dat	8.11e - 01 _{1.2e-02}	8.23e - 01 _{9.8e-03}	8.24e - 01 _{8.7e-03}
KC20-2fl-4rl.dat	8.50e - 01 _{2.7e-02}	8.65e - 01 _{2.0e-02}	8.73e - 01 _{2.2e-02}
KC20-2fl-5rl.dat	8.14e - 01 _{2.3e-02}	8.35e - 01 _{2.5e-02}	8.47e - 01 _{1.8e-02}
KC30-3fl-1uni.dat	5.61e - 01 _{1.7e-02}	5.83e - 01 _{1.5e-02}	5.73e - 01 _{1.5e-02}
KC30-3fl-1rl.dat	6.79e - 01 _{2.0e-02}	6.89e - 01 _{1.5e-02}	6.88e - 01 _{1.6e-02}
KC30-3fl-2uni.dat	4.94e - 01 _{3.9e-02}	5.33e - 01 _{4.3e-02}	5.32e - 01 _{4.2e-02}
KC30-3fl-2rl.dat	7.12e - 01 _{2.8e-02}	7.35e - 01 _{1.9e-02}	7.37e - 01 _{2.5e-02}
KC30-3fl-3uni.dat	5.50e - 01 _{1.1e-02}	5.56e - 01 _{1.1e-02}	5.57e - 01 _{9.5e-03}
KC30-3fl-3rl.dat	6.82e - 01 _{1.5e-02}	6.87e - 01 _{1.5e-02}	6.92e - 01 _{1.0e-02}

A Tabela 3 contém os resultados do indicador Hipervolume para diferentes taxas de cruzamento. A taxa de 50% obteve os melhores resultados para dez instâncias e foi, então, considerada para os testes seguintes.

6.2. Efeito da Seleção Automática dos Operadores feita pela HH

A Tabela 4 apresenta os resultados do indicador Hipervolume para a comparação do efeito da aplicação da Hiper-Heurística, em relação aos operadores PMX, OX e CX. Na tabela o valores em cinza escuro representam os melhores resultados e os valores em cinza claro indicam equivalência estatística de acordo com o teste Mann-Whitney-Wilcoxon [Conover 1999] com 95% de confiança. O algoritmo proposto obteve resultados melhores para seis instâncias e resultados equivalentes ao melhor para doze instâncias. O

Tabela 3. Cruzamento - Média e desvio padrão para o Hipervolume.

	30%	50%	75%
KC10-2ff-1uni.dat	8.44e - 011.2e-02	8.44e - 019.3e-03	8.40e - 011.2e-02
KC10-2ff-1rl.dat	7.80e - 012.5e-03	7.79e - 012.5e-03	7.79e - 013.0e-03
KC10-2ff-2uni.dat	9.80e - 011.1e-01	1.00e + 000.0e+00	9.77e - 011.2e-01
KC10-2ff-2rl.dat	9.21e - 017.1e-03	9.21e - 017.0e-03	9.22e - 015.8e-03
KC10-2ff-3uni.dat	6.26e - 011.9e-03	6.26e - 012.1e-03	6.26e - 012.1e-03
KC10-2ff-3rl.dat	8.27e - 017.1e-03	8.29e - 012.6e-03	8.26e - 017.1e-03
KC10-2ff-4rl.dat	7.39e - 018.0e-04	7.39e - 018.0e-04	7.39e - 011.0e-03
KC10-2ff-5rl.dat	9.38e - 012.7e-03	9.38e - 012.7e-03	9.38e - 012.7e-03
KC20-2ff-1uni.dat	7.43e - 011.7e-02	7.46e - 011.9e-02	7.39e - 011.8e-02
KC20-2ff-1rl.dat	8.00e - 012.4e-02	8.06e - 011.8e-02	7.95e - 011.8e-02
KC20-2ff-2uni.dat	6.32e - 016.1e-02	6.74e - 019.7e-02	6.36e - 017.5e-02
KC20-2ff-2rl.dat	8.37e - 012.2e-02	8.40e - 012.3e-02	8.38e - 011.9e-02
KC20-2ff-3uni.dat	6.17e - 014.6e-03	6.18e - 015.5e-03	6.13e - 017.5e-03
KC20-2ff-3rl.dat	8.24e - 018.6e-03	8.27e - 017.1e-03	8.23e - 019.8e-03
KC20-2ff-4rl.dat	8.91e - 011.4e-02	8.91e - 011.9e-02	8.90e - 011.3e-02
KC20-2ff-5rl.dat	8.37e - 011.6e-02	8.35e - 012.0e-02	8.39e - 011.3e-02
KC30-3ff-1uni.dat	5.82e - 011.5e-02	5.61e - 011.3e-02	5.50e - 011.7e-02
KC30-3ff-1rl.dat	6.87e - 011.1e-02	6.79e - 011.6e-02	6.59e - 011.7e-02
KC30-3ff-2uni.dat	5.62e - 012.9e-02	5.55e - 012.9e-02	5.30e - 013.5e-02
KC30-3ff-2rl.dat	7.36e - 012.3e-02	7.27e - 012.6e-02	7.09e - 012.8e-02
KC30-3ff-3uni.dat	5.84e - 018.6e-03	5.75e - 011.4e-02	5.59e - 011.2e-02
KC30-3ff-3rl.dat	7.02e - 011.4e-02	6.91e - 011.2e-02	6.87e - 011.2e-02

Tabela 4. Efeito do MOEA/D-DRA_{MAB} - Média e desvio padrão para o Hipervolume.

	MOEA/D-DRA _{MAB}	MOEA/D-DRA-PMX	MOEA/D-DRA-OX	MOEA/D-DRA-CX
KC10-2ff-1uni.dat	8.62e - 018.2e-03	8.63e - 018.7e-03	8.63e - 019.5e-03	8.59e - 011.1e-02
KC10-2ff-1rl.dat	7.85e - 012.5e-03	7.84e - 012.7e-03	7.85e - 012.6e-03	7.84e - 012.8e-03
KC10-2ff-2uni.dat	1.00e + 000.0e+00	9.76e - 011.3e-01	8.95e - 012.7e-01	8.75e - 012.8e-01
KC10-2ff-2rl.dat	9.21e - 017.0e-03	9.24e - 014.9e-03	9.23e - 014.6e-03	9.24e - 015.6e-03
KC10-2ff-3uni.dat	6.37e - 012.4e-03	6.37e - 012.0e-03	6.37e - 015.6e-04	6.37e - 011.6e-03
KC10-2ff-3rl.dat	8.29e - 012.6e-03	8.27e - 016.8e-03	8.27e - 015.6e-03	8.25e - 017.2e-03
KC10-2ff-4rl.dat	7.39e - 018.0e-04	7.39e - 011.8e-05	7.39e - 011.0e-03	7.38e - 012.1e-03
KC10-2ff-5rl.dat	9.38e - 012.7e-03	9.38e - 012.7e-03	9.38e - 012.7e-03	9.37e - 016.7e-03
KC20-2ff-1uni.dat	7.42e - 011.8e-02	7.38e - 011.3e-02	7.42e - 011.6e-02	7.46e - 011.6e-02
KC20-2ff-1rl.dat	8.13e - 011.7e-02	8.02e - 012.1e-02	8.09e - 011.7e-02	8.02e - 012.3e-02
KC20-2ff-2uni.dat	6.75e - 019.7e-02	6.54e - 017.4e-02	6.74e - 017.4e-02	6.51e - 018.4e-02
KC20-2ff-2rl.dat	8.34e - 012.3e-02	8.34e - 012.0e-02	8.28e - 011.7e-02	8.31e - 012.4e-02
KC20-2ff-3uni.dat	6.29e - 015.6e-03	6.28e - 015.1e-03	6.23e - 017.4e-03	6.29e - 015.1e-03
KC20-2ff-3rl.dat	8.22e - 017.1e-03	8.22e - 018.7e-03	8.14e - 016.6e-03	8.21e - 011.2e-02
KC20-2ff-4rl.dat	8.95e - 011.9e-02	8.97e - 011.3e-02	8.99e - 011.5e-02	8.94e - 011.7e-02
KC20-2ff-5rl.dat	8.55e - 011.9e-02	8.50e - 011.6e-02	8.49e - 011.9e-02	8.56e - 011.4e-02
KC30-3ff-1uni.dat	5.52e - 011.2e-02	5.56e - 011.2e-02	5.35e - 011.3e-02	5.71e - 011.2e-02
KC30-3ff-1rl.dat	6.76e - 011.6e-02	6.73e - 012.1e-02	6.45e - 011.5e-02	6.84e - 012.0e-02
KC30-3ff-2uni.dat	5.61e - 012.8e-02	5.67e - 013.1e-02	5.19e - 013.0e-02	5.79e - 013.6e-02
KC30-3ff-2rl.dat	7.27e - 012.7e-02	7.22e - 012.7e-02	6.99e - 012.4e-02	7.37e - 012.5e-02
KC30-3ff-3uni.dat	5.77e - 011.4e-02	5.81e - 011.0e-02	5.60e - 018.9e-03	5.88e - 011.2e-02
KC30-3ff-3rl.dat	6.93e - 011.2e-02	6.93e - 011.2e-02	6.79e - 011.0e-02	7.10e - 011.4e-02

MOEA/D-DRA_{MAB} só não foi capaz de se equiparar ao melhor algoritmo na solução de quatro instâncias com três objetivos. O operador CX apresentou os melhores resultados para dez instâncias. Os operadores PMX e OX apresentaram um desempenho inferior. A HH proposta obteve boas soluções tanto para instâncias com dois quanto para instâncias com três objetivos, sendo capaz de balancear as escolhas dos operadores.

7. Conclusões e Trabalhos Futuros

Este trabalho apresentou o desenvolvimento e análise de uma Hiper-Heurística de seleção para o Problema Quadrático de Alocação multiobjetivo. O algoritmo proposto, o MOEA/D-DRA_{MAB}, foi testado em 22 instâncias do *benchmark* de Knowles & Corne

que consideram diferentes quantidades de instalações/localidades e objetivos. Os experimentos foram conduzidos em duas etapas e levaram em consideração o indicador de qualidade Hipervolume. Na primeira etapa os seguintes parâmetros do algoritmo MOEA/D- DRA_{MAB} foram empiricamente ajustados. No segundo grupo de simulações foi analisada a capacidade da Hiper-Heurística durante a escolha automática dos operadores. Neste conjunto de testes o algoritmo proposto obteve desempenho satisfatório, obtendo o melhor resultado ou sendo equivalente ao melhor algoritmo na maioria das instâncias. Salienta-se que a Hiper-Heurística possui a vantagem de não necessitar de ajuste dos operadores.

Os resultados são promissores e foram identificados pontos a serem explorados em pesquisas futuras, tais como: o uso de buscas locais, o uso de um arquivo externo para o armazenamento das soluções não dominadas, diferentes funções de agregação.

Agradecimentos

Os autores agradecem ao CNPq e à Fundação Araucária pelo apoio financeiro.

Referências

- Auer, P. (2003). Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422.
- Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256.
- Bilgin, B., Özcan, E., and Korkmaz, E. E. (2006). An experimental study on hyper-heuristics and exam timetabling. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 394–412. Springer.
- Burke, E. K., Gendreau, M., Hyde, M. R., Kendall, G., Ochoa, G., Özcan, E., and Qu, R. (2013). Hyper-heuristics: a survey of the state of the art. *JORS*, 64(12):1695–1724.
- Ceberio, J., Calvo, B., Mendiburu, A., and Lozano, J. A. (2015). Multi-objectivising the quadratic assignment problem by means of an elementary landscape decomposition. In *Conference of the Spanish Association for Artificial Intelligence*, pages 289–300. Springer.
- Chakhlevitch, K. and Cowling, P. (2008). *Hyperheuristics: Recent Developments*, pages 3–29. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Coello, C. C., Lamont, G. B., and Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems*. Springer Science & Business Media.
- Conover, W. J. (1999). *Practical Nonparametric Statistics*. Wiley, 3 edition.
- de Almeida, C. P., Gonçalves, R. A., Goldberg, E. F. G., Goldberg, M. C., and Delgado, M. R. (2014). Transgenetic algorithms for the multi-objective quadratic assignment problem. In *BRACIS*, pages 312–317. IEEE.
- Drugan, M. M. (2014). Multi-objective quadratic assignment problem instances generator with a known optimum solution. In *International Conference on Parallel Problem Solving from Nature*, pages 559–568. Springer.
- Gonçalves, R. A., Almeida, C. P., and Pozo, A. (2015). Upper confidence bound (ucb) algorithms for adaptive operator selection in moea/d. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 411–425. Springer.

- Hardwick, J., Stout, Q. F., et al. (1991). Bandit strategies for ethical sequential allocation. *Computing Science and Statistics*, 23(421-424):6–1.
- Knowles, J. and Corne, D. (2002). Towards Landscape Analyses to Inform the Design of Hybrid Local Search for the Multiobjective Quadratic Assignment Problem. In Abraham, A., del Solar, J. R., and Koppen, M., editors, *Soft Computing Systems: Design, Management and Applications*, pages 271–279, Amsterdam. IOS Press. ISBN 1-58603-297-6.
- Knowles, J. and Corne, D. (2003). Instance generators and test suites for the multiobjective quadratic assignment problem. In Fonseca, C., Fleming, P., Zitzler, E., Deb, K., and Thiele, L., editors, *Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, April 2003, Proceedings*, number 2632 in LNCS, pages 295–310. Springer.
- Koopmans, T. C. and Beckmann, M. J. (1957). Assignment problems and the location of economics activities. *Econometrica*, 25(1):53–76.
- Larrañaga, P., Kuijpers, C., Murga, R., Inza, I., and Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2):129–170.
- Li, H. and Zhang, Q. (2009). Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302.
- Li, K., Fialho, A., Kwong, S., and Zhang, Q. (2014). Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 18(1):114–130.
- Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., and Querido, T. (2007). A survey for the quadratic assignment problem. *European journal of operational research*, 176(2):657–690.
- Robbins, H. (1985). Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer.
- Sahni, S. and Gonzales, T. (1976). P-complete approximation problems. *Journal of the Association for Computing Machinery*, 23(3):555–565.
- Umbarkar, A. and Sheth, P. (2015). Crossover operators in genetic algorithms: a review. *ICTACT Journal on Soft Computing*, 6(1):1083–1092.
- Vermorel, J. and Mohri, M. (2005). Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*, pages 437–448. Springer.
- Zhang, Q., Liu, W., and Li, H. (2009). The performance of a new version of moea/d on cec09 unconstrained mop test instances. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 203–208. IEEE.
- Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. (2011). Multi-objective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32 – 49.

Redes Neurais Convolucionais Aplicadas ao Processo de Classificação de Cultivares de Guaranazeiros

Alex de Lima Sousa¹, Marcos Filipe Alves Salame²,
Firmino José do Nascimento Filho³, André Luiz Atroch³

¹ Bolsista de Iniciação Científica FAPEAM, ² Analista, ³ Pesquisadores
Embrapa Amazônia Ocidental
Caixa Postal 319 – 69.010-970 – Manaus – AM – Brasil

allexlima@unn.edu.br,
{marcos.salame, firmino.filho, andre.atroch}@embrapa.br

Abstract. *The guarana holds important social and economic value for the Amazon due to its wide use in beverage and cosmetic industries. However, its vulnerability to local fungi and diseases causes a low productivity in the region, which has motivated the creation and availability of cultivars resistant to the main planting impediments, by Embrapa. The distinction of these genetic varieties, however, still depends on manual and technical resources, which are subject to human error. Thus, this work presents a specialized computational agent in the classification of some these cultivars that obtained a accuracy around 97% with a Convolutional Neural Networks model using an own image database.*

Resumo. *O guaranazeiro detém importante valor social e econômico para a Amazônia devido a sua ampla utilização nas indústrias de bebidas e cosméticos. Contudo, sua vulnerabilidade a fungos e doenças locais ocasiona uma baixa produtividade na região, o que tem motivado a criação e disponibilização de cultivares resistentes aos principais empecilhos de plantio, pela Embrapa. A distinção dessas variedades genéticas, entretanto, ainda depende de recursos manuais e técnicos, passíveis de falha humana. Assim, este trabalho apresenta um agente computacional especialista na classificação de alguns desses cultivares que, com o uso de Redes Neurais Convolucionais, obteve uma acurácia em torno de 97% utilizando uma base de imagens própria.*

1. Introdução

O fruto do guaranazeiro (*Paullinia cupana* var. *sorbilis*), o guaraná, devido as suas propriedades energéticas e medicinais, é considerado um importante insumo para indústrias de bebidas e cosméticos, agregando considerável valor socioeconômico para estados que o cultivam, como o Amazonas [Tavares et al. 2005].

Mesmo sendo uma planta originalmente amazônica, a maior concentração espacial de cultivo e produção brasileira do guaraná está na Bahia [IBGE 2015]. Um dos principais motivos que afetam o cultivo de guaranazeiros na Amazônia e incitaram essa migração foi a ausência de uma suficiente diversidade genética resistente a proliferação de pragas e doenças como a antracnose, causada pelo fungo *Colletotrichum guaranicola* [Nascimento Filho et al. 2007].

Esse cenário motivou, no final da década de 70, a Empresa Brasileira de Pesquisa Agropecuária (Embrapa), pela unidade Amazônia Ocidental, iniciar um programa de melhoramento genético do guaranzeiro que objetivava a expansão da produção do guaraná na Amazônia através de pesquisas que, desde 1996, desenvolveram e disponibilizaram 19 cultivares resistentes aos principais fatores bióticos e abióticos que afetam a guaranaicultura na região amazônica [Tricaud et al. 2016].

A quantidade de espécimes desenvolvidos e a variação fenotípica que cada cultivar apresenta ocasionou a elaboração de uma metodologia para distinção desses organismos a partir dos procedimentos esclarecidos no manual intitulado "Instruções para Execução dos Ensaio de Distingibilidade, Homogeneidade e Estabilidade de Cultivares de Guaraná (*Paullinia cupana* var. *Sorbilis*)", disponibilizado pelo Ministério da Agricultura, Pecuária e Abastecimento do Brasil em 2010 [MAPA 2010].

Os dispostos que esse manual fornece, entretanto, são restritos a informações técnicas que podem dificultar o processo de distinção dos cultivares caso quem esteja procedendo a análise não tenha alguma familiaridade com botânica ou cultivo do guaraná.

Assim, aliar pesquisa e desenvolvimento de soluções que envolvam a tecnologia da informação, elaborando novas ferramentas que possam auxiliar ou substituir processos difíceis e ainda manuais, como a distinção dos cultivares de guaranzeiros, são alternativas que podem colaborar com produtividade do guaraná no Estado do Amazonas, a médio ou longo prazo.

A utilização técnicas de aprendizado de máquinas para a classificação de plantas a partir de descritores morfológicos como folhas, por exemplo, possuem considerável abundância na literatura em consequência da contribuição importante que o reconhecimento de espécimes vegetais, por sistemas computacionais, oferece ao setor agrônomo e computacional.

Redes Neurais Convolucionais (ou, em inglês, *Convolutional Neural Networks* – CNNs) são consideradas um modelo matemático de classificação inspirado no sistema biológico de processamento de dados visuais. Na literatura, esses algoritmos dispõem de resultados satisfatórios para uma diversidade de problemas, incluindo classificação vegetal [Gu et al. 2015, Jassmann et al. 2015].

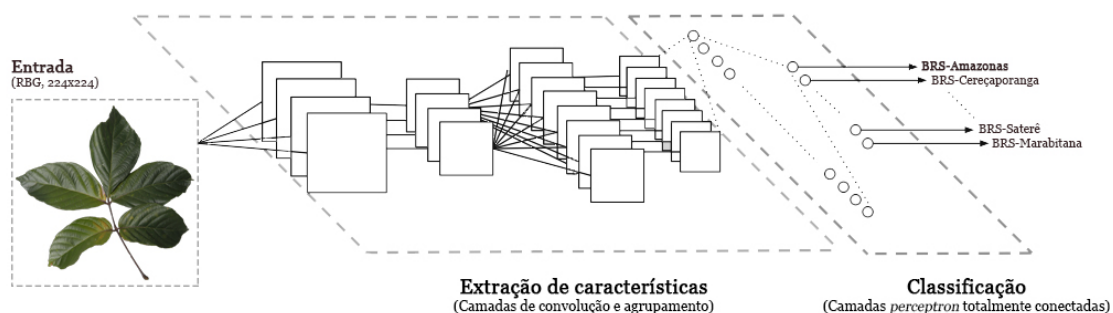


Figura 1. Modelo de Rede Neural Artificial Convolucional *LeNet* e suas fases de processamento ilustrando o processo de classificação de cultivares de guaraná. Adaptado de [Vargas et al. 2016].

Diferentemente das redes neurais convencionais, as camadas convolucionais das CNNs, conforme é ilustrado na Figura 1, possuem a responsabilidade de aplicar diversos filtros no dado de entrada, de forma a constituir uma espécie de hierarquia de características que resultam em descritores da imagem precisos [Krizhevsky et al. 2012, Ciresan et al. 2011].

Em contextos de classificação, após as camadas de responsáveis pela extração de características, geralmente é adicionada ao menos uma camada de neurônios artificiais totalmente conectados. Essa fase do processamento, em conjunto com uma função de ativação, é responsável por definir uma conexão entre os filtros executados nas primeiras camadas da rede e a classe do objeto analisado.

Nos trabalhos de [He and Tian 2016] e [Reyes et al. 2015], CNNs são empregadas ao processo de classificação de imagens foliares de diversos espécimes do *dataset Pl@ntView*. O primeiro trabalho conseguiu obter uma acurácia de 86.1% enquanto que o segundo, embora tenha utilizado um modelo pré-treinado com o *dataset ImageNet*, obteve aproximadamente 60% de acurácia. Ambos os trabalhos fizeram uso de imagens com fundos naturais. Provavelmente, essas amostras não tiveram o fundo segmentado devido ao grande volume de imagens nos *datasets* utilizados.

Algumas técnicas de aprendizado de máquina foram aplicadas ao contexto de classificação dos cultivares de guaranazeiros desenvolvidos pela Embrapa, onde observou-se que técnicas de aprendizado profundo, como redes neurais convolucionais, obtiveram uma melhor acurácia em comparação com técnicas clássicas como as árvores de decisão e máquinas de vetores de suporte.

Dessarte, propõe-se a elaboração de uma solução especialista no processo de classificação dos cultivares de guaranazeiros a partir da imagem de uma amostra foliar, utilizando CNNs. Ansiando, assim, a disponibilização de uma ferramenta com fácil utilização, confiável e ágil para auxiliar agricultores e pesquisadores que trabalham com a produção de guaraná na Amazônia, oferecendo uma alternativa às vias atuais de discriminação dos cultivares de guaranazeiros.

2. Material e Métodos

À vista de que os algoritmos de visão computacional e aprendizado de máquina, como as CNNs, podem ter custos computacionais elevados, optou-se por não fazer o emprego desses recursos diretamente em *smartphones*. Assim, almejando também uma arquitetura escalável para a solução proposta, foi-se estabelecido um fluxo para o sistema, conforme Figura 2, tendo como base a metodologia *RESTful*.

2.1. Interface de interação com o usuário e comunicação com o servidor

Um simples *webapp* ficou responsável por fazer o envio da imagem de uma folha do guaranazeiro para o servidor. Esse sistema foi desenvolvido visando ser compatível com dispositivos de diversos sistemas operacionais e tamanhos de telas, a partir de linguagens de marcação *web* com um *back-end* escrito em *javascript*.

Após o usuário capturar ou selecionar, a partir da galeria de imagens, a foto de uma amostra foliar, a aplicação se encarrega de iniciar uma comunicação *ajax* e enviar a imagem codificada em *base64* para o servidor, em uma requisição *POST* formatada com o padrão *json*.

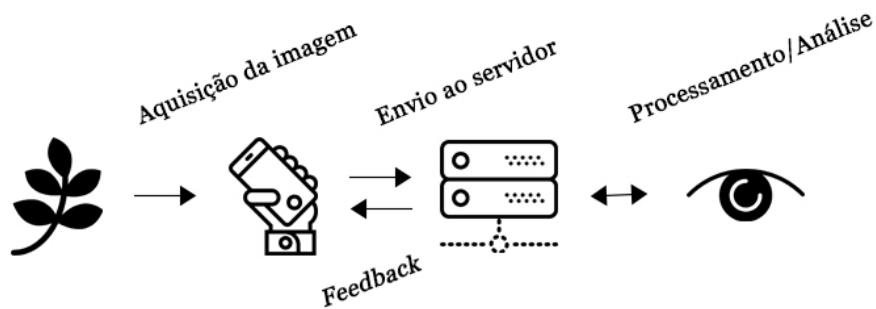


Figura 2. Arquitetura proposta: Após a aquisição da foto, por um dispositivo móvel, a imagem da folha é enviada para um servidor responsável pela classificação e *feedback* da análise.

As requisições ao servidor são gerenciadas por um *RESTful webservice* implementado através da linguagem de programação *Python* em conjunto com as bibliotecas *Flask* e *Flask-RESTful*. No servidor, o sistema foi dividido em dois principais módulos: i) o *webservice*, em si; e ii) o de classificação, incumbido de executar os processos de especificação da imagem submetida, através das CNNs.

2.2. Elaboração do *dataset* e pré-processamento gráfico

Ainda é comum a ausência de estudos relacionados a aprendizagem de máquinas e visão computacional com algumas plantas nativas da Amazônia, como guaranazeiro. Em consequência, bases de imagens dessa vegetação possuem pouca, ou nenhuma, presença em *datasets* públicos como *Pl@ntView*, *Flavia* e o *ImageNet*.

Dessa forma, foi-se necessário a criação de um *dataset* específico para este projeto, com imagens de amostras foliares adultas dos principais cultivares de guaranazeiro. Por ser uma espécie arbustiva e com folhas compostas não tão abundantes, apenas 40 amostras de dois cultivares, no total, foram coletadas em Janeiro e Março de 2017, no campo de guaranazeiros da Embrapa Amazônia Ocidental, na zona rural de Manaus ($2^{\circ}53'39.5''S$ $59^{\circ}58'20.8''W$).

Os cultivares selecionados foram o BRS-Amazonas e BRS-Cereçaporanga, ambos resistentes às principais doenças que atacam guaranazeiros convencionais [Nascimento Filho et al. 1999, Nascimento Filho et al. 2007]. As imagens foram capturadas no mesmo dia de cada coleta, em alta resolução. No mês de Janeiro, foram obtidas 20 fotos, uma de cada amostra, a partir do sensor *Canon EOS 6D*; e em Março, as imagens foram capturadas por um sensor *Nikon D7100* em 2 ângulos diferentes, totalizando 40 fotos, nesse mês. Em seguida, as imagens foram organizadas conforme suas classes (BRS-Amazonas e BRS-Cereçaporanga) e validadas por engenheiros agrônomos e pesquisadores especialistas no estudo de guaranazeiros, colaboradores deste trabalho.

O pré-processamento nas imagens baseou-se nos métodos aplicados em trabalhos como os de [Krizhevsky et al. 2012], [Simonyan and Zisserman 2014] e [Reyes et al. 2015]. Assim, cada uma das 60 imagens foram redimensionadas para 224×224 *pixels*, com a folha centralizada no escopo de canais RGB e, diferentemente desses estudos citados, foi possível, devido à pouca quantidade de imagens, realizar a segmentação dos fundos onde as fotografias foram tiradas.

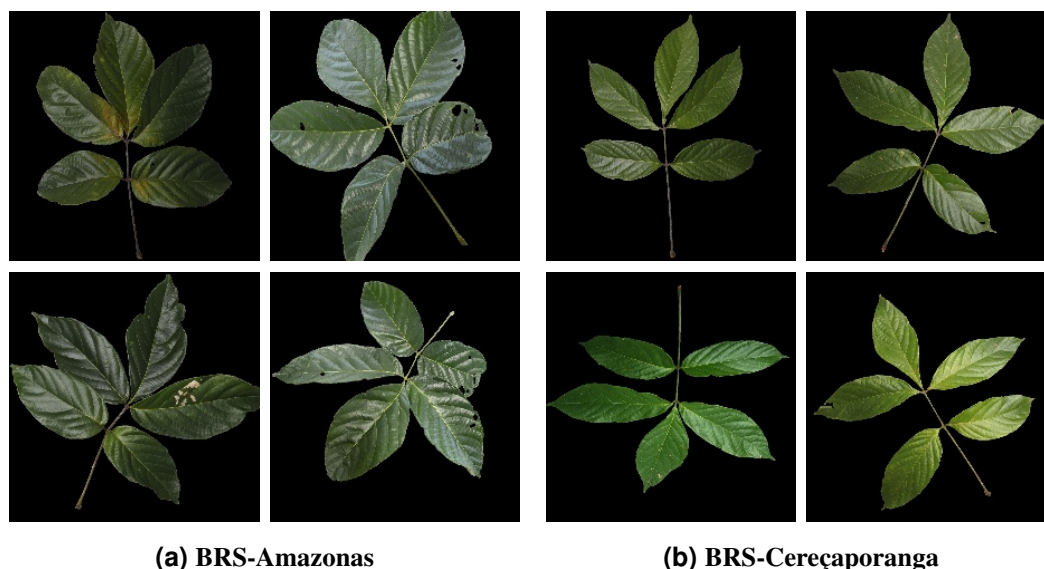


Figura 3. Amostras de imagens presentes no *dataset* elaborado com as duas classes utilizadas no experimento, após o pré-processamento e *data augmentation*.

Com um *dataset* relativamente pequeno, quando comparado a outros trabalhos relacionados, optou-se por aplicar um processo conhecido por *data augmentation*: um recurso amplamente empregado para o avolumamento de *datasets* limitados a uma pequena quantidade de dados e que, também, pode auxiliar a redução de *overfitting*, um dos principais problemas que podem ocorrer durante o treinamento de um classificador [Krizhevsky et al. 2012].

Tabela 1. Distribuição de imagens do *dataset*, gerada aleatoriamente, utilizada no treino e avaliação das arquiteturas de classificação selecionadas.

	Treino		Avaliação		Total
	AlexNet	LeNet	AlexNet	LeNet	
BRS-Amazonas	928	925	472	475	1400
BRS-Cereçaporanga	948	951	452	449	1400
Total	1876 (67%)		924 (33%)		2800

A biblioteca *Keras*, por meio do seu módulo de processamento de imagens, permite a execução da técnica de *data augmentation* criando novas imagens sintéticas a partir de variações randômicas de cada uma das fotografias presentes no *dataset* [Chollet 2015]. Dessa forma, essa biblioteca foi utilizada combinada com os seguintes parâmetros: i) ângulo de 35° para o intervalo de rotação; ii) espelhamento vertical e horizontal; iii) modo de preenchimento constante; iv) 0.01 de o intervalo de zoom; v) 0.2 para a taxa de variação dos canais de cor; e vi) $cval = 0$.

Em consequência, esse processo resultou uma ampliação de, aproximadamente, 46.7 vezes o volume inicial de imagens do *dataset*, com imagens organizadas randomica-

mente seguindo a metodologia *holdout*, conforme explana a Tabela 1. Algumas amostras resultantes da técnica de *data augmentation* são apresentadas na Figura 3.

2.3. Escopo de classificação

Dentre os diversos modelos de implementação das CNNs existentes, a arquitetura *LeNet*, ilustrada na Figura 1, foi introduzida na literatura pelo trabalho de [LeCun et al. 1998] e é considerada um dos primeiros modelos de redes neurais convolucionais, além de possuir uma estrutura simples e pequena quando comparada com arquiteturas mais atuais como a proposta por [Krizhevsky et al. 2012], a *AlexNet*, e a VGG-16.

Dentre os diversos modelos e configurações de treino experimentadas previamente, observou-se que a arquitetura VGG-16 não conseguiu alcançar uma performance satisfatória, obtendo uma acurácia 20% a menos que os modelos *LeNet* e *AlexNet*. Devido a esse fator e à sólida profusão na literatura, as arquiteturas apresentadas nos trabalhos [LeCun et al. 1998] e [Krizhevsky et al. 2012] foram selecionadas para serem aplicadas e avaliadas no contexto de classificação de amostras foliares de cultivares do guaranazeiro.

A implementação desses modelos foi realizada com a linguagem de programação *Python*, por meio da biblioteca de *deep learning Keras*, utilizando o *back-end TensorFlow*, além de outras como *NumPy*, *Scikit-learn* e *Scikit-image*. Cada um dos classificadores foram treinados com os seguintes parâmetros: i) 100 épocas de treino; ii) uma taxa de aprendizagem igual a 0.01; e iii) aproximadamente, 1880 imagens, conforme Tabela 1.

O ambiente utilizado para realizar tanto o treinamento quanto as avaliações foi uma instância dedicada no *Google Cloud Platform* com processamento *Intel® Xeon®* de 16 núcleos com 2,6GHz cada, 30GBi de memória RAM e sistema operacional *GNU/Linux Debian*.

3. Resultados e Discussão

O sistema final foi nomeado como WAVA (*Waraná's Automated Visual Analysis*) e apresentou resultados promissores, de forma geral. É possível observar as interfaces gráficas de um protótipo funcional *web* da ferramenta na Figura 3, executado a partir de um dispositivo móvel. A tela presente na Figura 4a dispõe de uma estrutura que comporta o envio ou aquisição da foto de uma amostra foliar do guaranazeiro e a envia para o servidor, onde a análise é, então, realizada. O *feedback* relacionado a classificação da imagem enviada é apresentado pela tela contida na Figura 4b.

Em relação ao módulo de classificação desenvolvido, logo após os treinamentos foi possível realizar os devidos testes com 33% das imagens do *dataset* que resultaram as matrizes de confusão presentes na Figura 5. Essas matrizes são responsáveis por organizar o conjunto de informações classificadas corretamente em suas diagonais principais, conforme é destacado na Figura 6a e 6b, enquanto que os itens classificados de forma errônea são dispostos nas diagonais secundárias.

A matriz de confusão detém considerável relevância por possibilitar o cálculo de métricas como a acurácia, precisão, revocação (ou *recall*) e o *F-Score*, utilizadas comumente para a avaliar modelos de classificação. A Tabela 2, por sua vez, apresenta os resultados de algumas dessas métricas para ambas as arquiteturas de CNN.

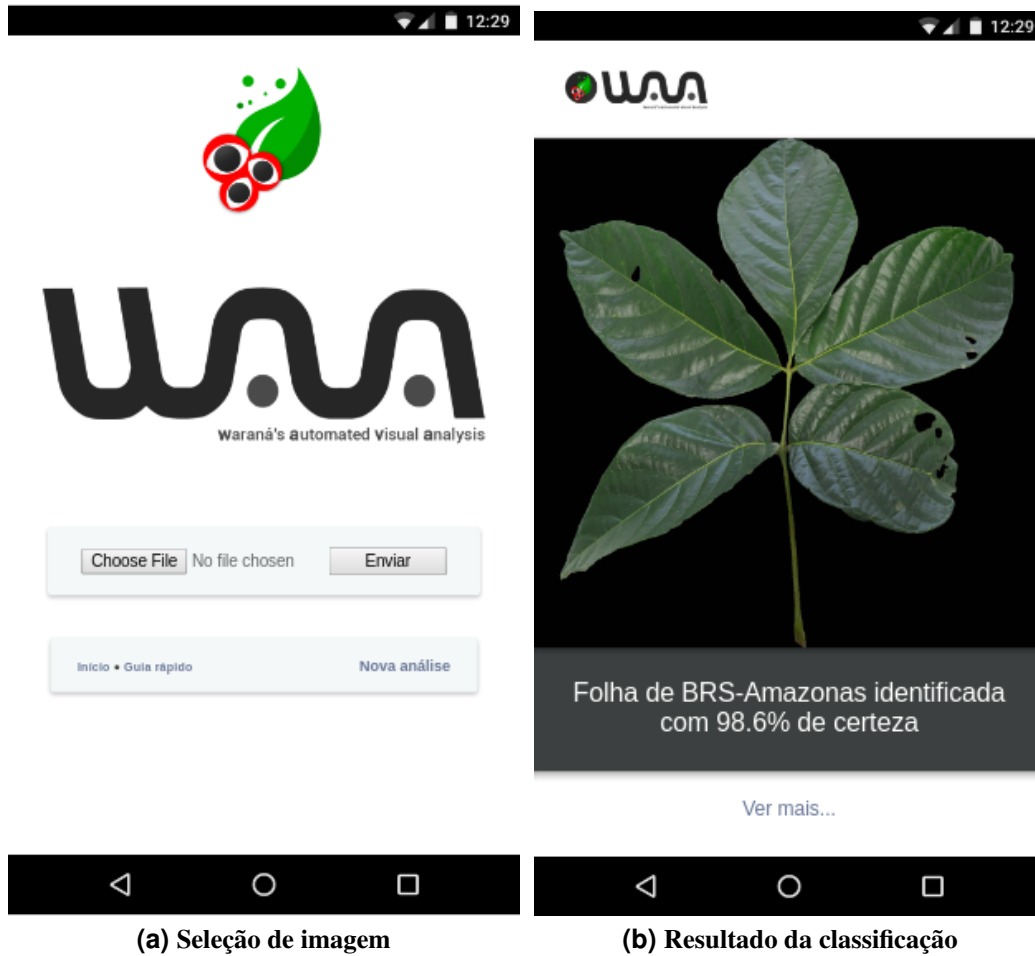


Figura 4. Telas do WebApp desenvolvido, demonstrando o reconhecimento de uma amostra de teste do dataset.

Figura 5. Matrizes de confusão das arquiteturas de classificação utilizadas.

		Classificado como	
		BRS-Amazonas	BRS-Cereçaporanga
Espécime	BRS-Amazonas	455	17
	BRS-Cereçaporanga	8	444

(a) AlexNet

		Classificado como	
		BRS-Amazonas	BRS-Cereçaporanga
Espécime	BRS-Amazonas	424	51
	BRS-Cereçaporanga	33	416

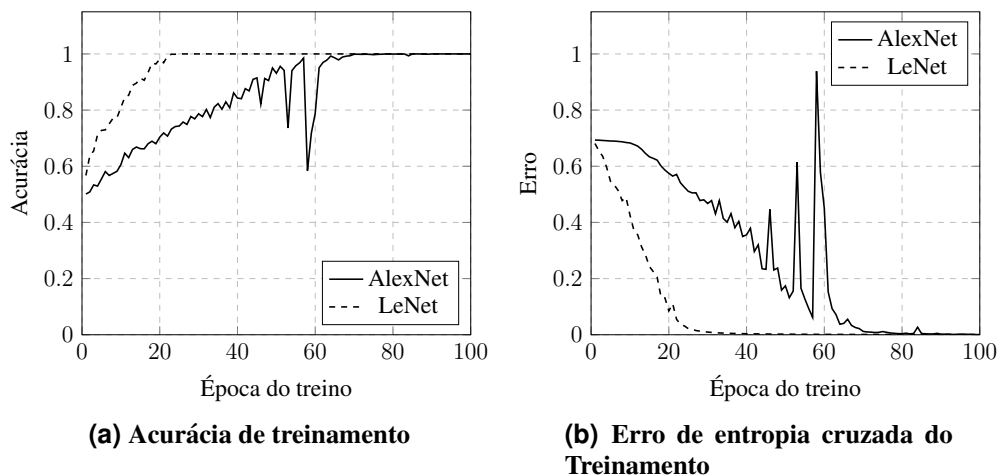
(b) LeNet

O comportamento em relação as taxas de acerto e erro de ambos os modelos utilizados durante o processo de treinamento é representado na Figura 6a e 6b, respectivamente, onde é possível observar uma certa dificuldade de convergência no modelo AlexNet, mesmo este obtendo uma acurácia de 97.3%, nas avaliações. Isso pode ter sido

Tabela 2. Métricas relacionadas a performance de ambos os classificadores.

	Precisão		Revocação		<i>F-Score</i>	
	AlexNet	LeNet	AlexNet	LeNet	AlexNet	LeNet
BRS-Amazonas	0.98	0.93	0.96	0.89	0.97	0.91
BRS-Cereçaporanga	0.96	0.89	0.98	0.93	0.97	0.91
Média	0.97	0.91	0.97	0.91	0.97	0.91

causado em consequência da "profundidade" e robustez que essa arquitetura apresenta, em relação à LeNet. Acredita-se, então, que aumentar a quantidade de épocas do treino desse modelo pode colaborar com uma melhor convergência. A rede LeNet, por outro, lado apresentou uma convergência bem mais ágil alcançando uma acurácia de 90.9%.

**Figura 6. Métricas de treinamento dos modelos de Redes Neurais Convolucionais AlexNet e LeNet.**

Entre as épocas 40 e 65 de treinamento do classificador AlexNet, é possível observar (vide Figura 6) anomalias que podem indicar a presença de *overfitting* neste modelo, mesmo obtendo as melhores métricas nos testes realizados, com uma acurácia de 97.3% contra os 90.9% do modelo LeNet.

4. Conclusões

Nas reuniões realizadas ao longo desse trabalho com pesquisadores que estudam os guarazeiros desenvolvidos pela Embrapa, foi-se observado a importância e as comodidades que a solução proposta neste trabalho pode oferecer, complementando os atuais métodos manuais de distinção dos cultivares.

O desenvolvimento do *dataset* utilizado, com apenas dois espécimes (BRS-Amazonas e BRS-Cereçaporanga) mostrou-se promissor, o que motiva sua ampliação, no futuro, de forma a incluir novas classes além de amostras foliares de mudas. Almeja-se também, distribuí-lo visando contribuir com possíveis trabalhos relacionados a guaranai-cultura, na área de computação e/ou botânica. Não obstante, será importante analisar

melhor o desempenho de novos testes, durante a ampliação do *dataset*, objetivando evitar a possibilidade real de *under* ou *overfitting*. Dessa forma, podendo, inclusive, explorar novas metodologias como a validação cruzada e *fine-tuning*.

Além da necessidade de uma análise mais profusa em relação à possibilidade de *overfitting* neste trabalho, algumas melhorias fazem-se necessárias em uma próxima versão do sistema WAVA como, por exemplo, a agregação de um pré-processamento automatizado que seja responsável pela subtração do *background* das imagens obtidas em ambientes não controlados para reduzir os ruídos para facilitar o processo de classificação.

Ressalta-se também a importância de uma validação, futuramente, em campos de guaranáis com agricultores e especialistas da guaranaicultura visando atestar a eficácia da ferramenta proposta neste trabalho que, até então, dispõe apenas de testes experimentais em ambientes controlados. Essa etapa precede as fases finais do projeto, junto da disponibilização de um aplicativo nativo para a plataforma *Google Android*. Com isso, espera-se lançar a solução visando auxiliar o público que atua com a guaranaicultura no Brasil.

5. Agradecimentos

À Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM), pela disponibilização da bolsa de pesquisa; à Empresa Brasileira de Pesquisa Agropecuária (Embrapa) – Amazônia Ocidental, pela infraestrutura disponibilizada; aos colaboradores deste trabalho, os pesquisadores e especialistas em guaranaicultura Firmino José do Nascimento Filho e André Luiz Atroch, cuja disponibilidade e conhecimentos técnicos foram imprescindíveis; e à toda a equipe do Núcleo de Tecnologia da Informação, em especial, ao Sr. Raimundo Barbosa e ao orientador deste projeto, Marcos Filipe Alves Salame.

Referências

- Chollet, F. (2015). Keras. <https://github.com/fchollet/keras>.
- Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L., and Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1237. Barcelona, Spain.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., and Wang, G. (2015). Recent advances in convolutional neural networks. *arXiv preprint arXiv:1512.07108*.
- He, A. and Tian, X. (2016). Multi-organ plant identification with multi-column deep convolutional neural networks. In *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, pages 002020–002025. IEEE.
- IBGE (2015). Pesquisa mensal de previsão e acompanhamento das safras agrícolas no ano civil. *Levantamento Sistemático da produção Agrícola*, 29(4):49.
- Jassmann, T. J., Tashakkori, R., and Parry, R. M. (2015). Leaf classification utilizing a convolutional neural network. In *SoutheastCon 2015*, pages 1–3. IEEE.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and

- Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- MAPA (2010). Instruções para execução dos ensaios de distinguibilidade, homogeneidade e estabilidade de cultivares de guaraná (*Paullinia cupana* var. *Sorbilis*). *República Federativa do Brasil: Diário Oficial da União*.
- Nascimento Filho, F. J., Atroch, A., Pereira, J., and Araújo, J. (2007). Brs-cereçaporanga: nova cultivar para o agronegócio do guaraná. *Embrapa Amazônia Ocidental. Comunicado Técnico*.
- Nascimento Filho, F. J., Atroch, A. L., da Silva Cravo, M., de Macêdo, J. L. V., Garcia, T. B., Júnior, R. C. C., and Ribeiro, J. d. R. C. (1999). *Clones de guaranazeiro para o Estado do Amazonas*. Embrapa Amazônia Ocidental.
- Reyes, A. K., Caicedo, J. C., and Camargo, J. E. (2015). Fine-tuning deep convolutional networks for plant recognition. In *CLEF (Working Notes)*.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.
- Tavares, A., Atroch, A., Nascimento Filho, F. J., Pereira, J., Araújo, J., Moraes, L., et al. (2005). Cultura do guaranazeiro no Amazonas. *Manaus: EMBRAPA Amazônia Ocidental. Sistema de Produção*, 4.
- Tricaud, S., Pinton, F., and Pereira, H. d. S. (2016). Saberes e práticas locais dos produtores de guaraná (*paullinia cupana* kunth var. *sorbilis*) do médio Amazonas: duas organizações locais frente à inovação. *Boletim do Museu Paraense Emílio Goeldi. Ciências Humanas*, 11(1):33–53.
- Vargas, A. C. G., Paes, A., and Vasconcelos, C. N. (2016). Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. *Conference on Graphics, Patterns and Images (SIBGRAPI)*, 29.

Combinando novidade e popularidade em recomendações personalizadas através do aprendizado do perfil do usuário

Ricardo M. Bertani¹, Reinaldo A. C. Bianchi², Anna H. R. Costa¹

¹ Laboratório de Técnicas Inteligentes – Universidade de São Paulo (EP/USP)
Av. Prof. Luciano Gualberto, 158 - trav. 3
05508-010 - São Paulo - SP - Brasil

² Departamento de Engenharia Elétrica - Centro Universitário FEI
Av. Humberto A. C. Branco, 3972
09850-901 - São Bernardo do Campo - SP - Brasil

ricardo.bertani@usp.br, rbianchi@fei.edu.br, anna.reali@usp.br

Abstract. *Recommender systems have been widely used to help users search for relevant content according to their preferences. A wide variety of algorithms have been proposed for this purpose, in particular those based on collaborative filtering, which still fail in sparse data, affecting the quality of recommendations. In order to mitigate this gap, a hybrid algorithm based on diffusion has been proposed in the literature. However, it does not differentiate users according to their profiles. In this paper, a new algorithm is presented for learning user profiles and consequent generation of personalized recommendations through diffusion, combining novelty with popular items. The experiments were performed in two sets of data and show that the results outperform those obtained by the original diffusion-based algorithm, in the same settings.*

Resumo. *Sistemas de recomendação têm sido amplamente utilizados para ajudar os usuários a procurar conteúdo relevante de acordo com suas preferências. Uma ampla variedade de algoritmos foi proposta para esse fim; em particular aqueles baseados em filtragem colaborativa, que ainda falham em dados esparsos, afetando a qualidade das recomendações. Para mitigar essa lacuna, um algoritmo híbrido baseado em difusão foi proposto na literatura; no entanto, este algoritmo não diferencia os usuários de acordo com seus perfis. Neste artigo, um novo algoritmo é apresentado para a aprendizagem do perfil dos usuários e consequente geração de recomendações personalizadas através da difusão, combinando novidades a itens populares. Os experimentos realizados em dois conjuntos de dados mostram que os resultados superam aqueles obtidos pelo algoritmo original baseado em difusão, nas mesmas configurações.*

1. Introdução

Sistemas de Recomendação (SR) são programas de computador utilizados para recomendar itens relevantes para usuários específicos através da predição de seus interesses particulares. Itens podem referir-se a filmes, músicas, notícias ou livros, por exemplo [Kotkov et al. 2016]. SR normalmente produzem uma lista de recomendação, a qual é ordenada por algum critério de relevância.

Diversas abordagens têm sido usadas em SR [Bobadilla et al. 2013, Lu et al. 2015b, Sanchez-Moreno et al. 2016, Lu et al. 2015a, Ricci et al. 2010], dentre as quais a mais popular consiste na filtragem colaborativa (FC), a qual explora informações prévias sobre o comportamento de uma comunidade de usuários para prever quais itens seriam de interesse do usuário alvo da recomendação [Jannach et al. 2010, Christakopoulou and Karypis 2016, Sanchez-Moreno et al. 2016, Kotkov et al. 2016].

Os algoritmos de FC podem ser classificados como baseados em memória ou baseados em modelo [Lü et al. 2012, Liu et al. 2014, Kotkov et al. 2016]. Na abordagem baseada em memória, a recomendação é feita primeiramente através da criação de um perfil para o usuário alvo da recomendação com base nas avaliações (*ratings*) que foram anteriormente atribuídas por ele aos itens, então são identificados aqueles usuários que avaliaram os mesmos itens de forma semelhante; finalmente, são recomendados ao usuário alvo os itens preferidos pelos usuários similares a ele. A avaliação mencionada trata-se de um valor utilizado para expressar o nível de interesse de um usuário por um item. Por outro lado, na abordagem baseada em modelo, há a construção de um modelo probabilístico (ou qualquer outro tipo) a partir de um grande conjunto de dados com avaliações atribuídas aos itens, e este modelo é então utilizado para prever as avaliações de itens desconhecidos e gerar as recomendações.

Neste artigo o interesse consiste na classe de algoritmos FC baseados em memória, os quais também são conhecidos na literatura como baseados em vizinhança (*neighborhood-based*). Nesta classe, os usuários mais similares ao usuário alvo da recomendação são considerados como vizinhos mais próximos [Bobadilla et al. 2013, Lu et al. 2015a, Christakopoulou and Karypis 2016, Sanchez-Moreno et al. 2016]. Para a determinação desta vizinhança de usuários, normalmente são consideradas algumas medidas de similaridade, sendo a mais comum o coeficiente de correlação de Pearson [Ricci et al. 2010], utilizado para medir o grau de correlação entre as avaliações de dois usuários, estabelecendo que aqueles usuários que avaliam os mesmos itens de forma similar são vizinhos.

Apesar de seu sucesso, a abordagem FC sofre com dois problemas conhecidos: partida a frio (*cold-start*) e esparsidade dos dados (*data sparsity*) [Shambour and Lu 2015, Choi et al. 2012]. O primeiro trata da dificuldade em recomendar de forma efetiva decorrente de uma falta inicial de avaliações devido tanto a novos itens quanto a novos usuários; já o último ocorre decorrente do alto número de itens no sistema, que torna praticamente impossível para os usuários avaliarem uma quantidade significativa de itens e, portanto, torna a vizinhança de usuários similares insuficiente, o que afeta diretamente a qualidade das recomendações.

Com relação ao problema da esparsidade, alguns algoritmos baseados em difusão (*diffusion-based*) foram propostos com o objetivo de suavizá-lo [Zhou et al. 2010, Zeng et al. 2013, Zhang and Zeng 2015, Chen et al. 2015]. Nestes algoritmos, itens e usuários são representados como vértices de um grafo bipartido, e cada interação entre eles (dada, por exemplo, por uma avaliação) define uma aresta, formando uma rede bastante complexa. A ideia consiste em explorar o fato de que opiniões expressas por um usuário particular podem ser propagadas por toda a rede. Entretanto, estas propagações normalmente são regidas por uma mesma sistematização, tornando as recomendações mais propensas a privilegiar ou itens populares ou novidades, sem considerar o gosto

particular de cada usuário.

Neste artigo é proposto um novo algoritmo de difusão que leva em consideração o perfil do usuário alvo para determinar sua preferência particular por itens populares ou por novidades. A proposta baseia-se na percepção de que usuários com perfis similares, geralmente, são propensos a se interessarem por itens em comum, sejam estes populares ou pouco conhecidos no sistema. Para isso, uma análise prévia de um conjunto de dados conhecidos é feita, visando agrupar usuários com perfis similares. Estes grupos definem um grau adequado de combinação entre popularidade e novidade dos itens a serem oferecidos na lista de itens recomendados. Assim, para cada usuário alvo do SR, o grau apropriado de combinação é definido de acordo com o seu perfil, influenciando o processo de difusão e gerando listas de recomendação personalizadas.

As seções seguintes estão organizadas da seguinte forma. A Seção 2 apresenta os métodos baseados em difusão, no qual a presente proposta se baseia. Na Seção 3 a proposta deste artigo é detalhada. Em seguida, na Seção 4 são descritas as métricas utilizadas para a avaliação da qualidade das recomendações geradas. Na Seção 5 são descritos os experimentos e discutidos os resultados obtidos. Finalmente, a Seção 6 apresenta as conclusões.

2. Métodos baseados em difusão

Na tentativa de contornar o problema da esparsidade de dados algumas abordagens baseadas em difusão têm obtido sucesso [Ricci et al. 2010]. Elas utilizam alguns processos físicos para explorar os relacionamentos indiretos entre usuários e itens dentro de um sistema. Em particular, um método com resultados bastante impressionantes foi o algoritmo híbrido baseado em difusão [Zhou et al. 2010] — a partir de agora aqui referenciado como *Hybrid* — que combina os algoritmos *Mass Diffusion(MD)* [Zhou et al. 2007] e *Heat-Spreading(HS)* [Zhang et al. 2007] para a geração de recomendações.

No *Hybrid*, um SR é representado por um grafo bipartido, definido formalmente como $G = \{U, I, E\}$, onde $U = \{u_1, u_2, \dots, u_N\}$, $I = \{i_1, i_2, \dots, i_M\}$ e $E = \{e_1, e_2, \dots, e_K\}$ são o conjunto de usuários, o conjunto de itens e o conjunto de arestas do grafo, respectivamente. Dado um conjunto de dados previamente conhecido, o grafo G é construído, atribuindo cada usuário a um vértice $u_i \in U$, cada item distinto a um vértice $i_j \in I$ e, caso o usuário u_i tenha tido alguma interação com o item i_j , uma aresta $e_s \in E$ é inserida em G , tornando u_i adjacente a i_j . Então, para cada usuário alvo da recomendação (u) presente na base de dados, são executadas as três etapas seguintes:

Passo 1 Primeiramente, um valor de recurso $r(u, i)$ é atribuído a cada item i no grafo, da seguinte forma: se existe uma aresta entre i e u então $r(u, i) = 1$, caso contrário, $r(u, i) = 0$.

Passo 2 Na segunda etapa, ocorre um processo de propagação dos valores de recurso do lado item para o lado usuário no grafo; cada item i com $r(u, i) > 0$ propaga um novo valor de recurso recalculado $r'(u, i)$ para cada usuário u com o qual é adjacente no grafo G .

Passo 3 Na etapa final do processo, ocorre uma nova propagação dos recursos do lado usuário para o lado item no grafo, na qual cada item i

recebe um novo valor de recurso recalculado $r''(u, i)$ a partir dos valores de recurso do lado usuário, $r'(u, i)$.

O cálculo dos novos valores de recurso nos Passos 2 e 3, representados respectivamente por $r'(u, i)$ e $r''(u, i)$, dependem do algoritmo considerado, MD, HS ou *Hybrid*, além da quantidade de interações que os usuários e itens tiveram na base de dados do SR, o que é representado pelo grau dos vértices no grafo, dado por $w(u)$ e $w(i)$, sendo respectivamente o grau do usuário u e o grau do item i . Logo, quando é utilizado puramente o algoritmo MD, os cálculos de $r'_{MD}(u, i)$ e $r''_{MD}(u, i)$ são:

$$r'_{MD}(u, i) = \sum_{i \in I} \frac{r(u, i)}{w(i)}; \quad e \quad (1)$$

$$r''_{MD}(u, i) = \sum_{u \in U} \frac{r'_{MD}(u, i)}{w(u)} \quad (2)$$

Por outro lado, quando é utilizado puramente o algoritmo HS, os cálculos de $r'_{HS}(u, i)$ e $r''_{HS}(u, i)$ são:

$$r'_{HS}(u, i) = \frac{\sum_{i \in I} r(u, i)}{w(i)}; \quad e \quad (3)$$

$$r''_{HS}(u, i) = \frac{\sum_{u \in U} r'_{HS}(u, i)}{w(u)} \quad (4)$$

Contudo, o algoritmo *Hybrid* [Zhou et al. 2010] ainda permite a combinação de ambos, MD e HS, através de um parâmetro de sincronização λ , o qual é utilizado para mesclar os cálculos das equações 1 e 3 em $r'_{HB}(u, i)$ (primeira etapa de propagação) e 2 e 4 em $r''_{HB}(u, i)$ (segunda etapa de propagação):

$$r'_{HB}(u, i) = \left(\sum_{i \in I} \frac{r(u, i)}{w(i)^\lambda} \right) / w(i)^{(1-\lambda)}; \quad e \quad (5)$$

$$r''_{HB}(u, i) = \left(\sum_{u \in U} \frac{r'_{HB}(u, i)}{w(u)^\lambda} \right) / w(u)^{(1-\lambda)}. \quad (6)$$

Observe que quando $\lambda = 0$ tem-se a aplicação original do algoritmo HS e, quando $\lambda = 1$, tem-se a aplicação original do algoritmo MD. Qualquer valor entre 0 e 1 permite a combinação de ambos no algoritmo *Hybrid*.

Após a execução dos três passos descritos anteriormente, referentes ao usuário alvo u , todos itens com um valor final de recurso positivo, $r''(u, i) > 0$, são então ordenados de forma crescente, resultando na chamada lista de predição. Os itens com maiores valores de recurso (posições finais da lista de predição) são então selecionadas para serem recomendados ao usuário, compondo a lista de recomendação. Na prática, esta lista possui um tamanho fixo definido previamente no SR e é ordenada de forma decrescente pelo valor de recurso dos itens, sendo que a primeira posição contém o item com maior valor

de recurso (predito como o mais significativo para o usuário alvo), e a última, o item com menor valor de recurso.

A grande particularidade dos métodos MD e HS se dá pelo fato de que, enquanto o primeiro tende a gerar recomendações compostas de itens mais populares (vértices do grafo com maiores graus), o segundo tende a gerar recomendações compostas de itens menos conhecidos (vértices do grafo com menores graus). Na literatura foram ainda apresentadas algumas variações do método *Hybrid*, em especial aquela conhecida como *Semi-Local Diffusion (SLD)* [Zeng et al. 2013], que permite realizar uma quantia maior de propagações de recursos no grafo bipartido do que o *Hybrid*. Na prática, isto significa que os passos 2 e 3 de propagação ocorrem mais do que uma vez e é possível alcançar vizinhanças maiores de usuários e itens.

Entretanto, observa-se que tanto no método *Hybrid* quanto no SLD, o valor do parâmetro λ sempre é o mesmo para todos usuários do SR, sem considerar qualquer informação sobre seus perfis. Na prática, os autores consideram $0.0 \leq \lambda \leq 1.0$, onde $\lambda \in \Lambda$ e $\Lambda = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ e calculam, para cada valor do conjunto Λ , as listas de recomendação. A seguir, usando um conjunto de dados de validação, escolhem experimentalmente o λ que maximiza alguma métrica escolhida. Este mesmo λ é utilizado para todo e qualquer usuário alvo do SR. Nossa proposta consiste em adaptar o valor de λ ao perfil de cada usuário alvo, e é detalhada na próxima seção.

3. Difusão orientada a Perfil de Usuário

A proposta do presente artigo consiste em um algoritmo que combina o método *Hybrid* com o aprendizado supervisionado do parâmetro λ para geração de recomendações personalizadas, através de difusão. Este método foi nomeado como “Difusão orientada a Perfil de Usuário”; ou *User Profile Oriented Diffusion (UPOD)*, em inglês, e vem melhorar o cálculo e aplicação do parâmetro λ nas recomendações efetuadas pelo SR. Vale lembrar que o mesmo procedimento pode ser expandido para o método SLD.

A ideia consiste em agrupar usuários que compartilhem perfis similares e definir, para cada tipo de perfil, o fator de combinação λ mais adequado para as difusões MD e HS no algoritmo *Hybrid*. Assim, cada usuário alvo terá o λ mais apropriado para seu perfil, combinando novidade e popularidade nas recomendações, de forma personalizada. O algoritmo é composto de duas fases: fase de treinamento e fase de recomendação, descritas a seguir.

3.1. Fase de Treinamento

A fase de treinamento tem por objetivo a construção de um classificador capaz de prever o valor mais adequado do fator de combinação λ em função dos perfis dos usuários, dados por atributos tais como: gênero, idade e localização. Outros atributos podem ser usados, desde que sejam disponibilizados nas bases de dados. O fator λ mescla difusões MD e HS no algoritmo *Hybrid*, combinando novidade e popularidade nas recomendações.

O algoritmo 1 detalha a fase de treinamento do UPOD, cujo objetivo é gerar um conjunto de treinamento (inicialmente vazio, no passo 1) para treinar um classificador. A partir do grafo G que representa as interações de usuários e itens de uma base de dados previamente conhecida, um conjunto de usuários de treinamento U_{train} é criado, contendo

os valores dos atributos $Feat$ para cada usuário de G (passo 2). Em seguida (passo 3), o conjunto de usuários U_{train} é agrupado em k grupos menores através de um algoritmo de agrupamento. No caso, foi utilizado o algoritmo k -Modes, uma extensão do algoritmo k -Means, voltado para o agrupamento de dados categóricos [Huang 1998]. Foram utilizados os atributos dos usuários para realizar o agrupamento, de forma que quanto mais valores de atributos em comum dois usuários tiverem, maior a chance de ficarem no mesmo grupo.

Na sequência, para cada grupo de usuários resultante (laço for, do passo 4 ao 10), aplica-se repetidamente o algoritmo *Hybrid* considerando cada um dos valores do conjunto Λ para o fator λ , $\lambda \in \Lambda$, e estima-se o melhor valor, λ_{best} , de acordo com os resultados obtidos pela métrica escolhida M (passo 5). Então, para cada usuário de cada um dos k grupos, são obtidos os valores de seus atributos (passo 7) e é gerado o par $\langle \text{ValorAtributos}, \lambda_{best} \rangle$ (passo 8), usado posteriormente (no passo 11) para treinar um classificador. No final deste processo, tem-se o classificador treinado a partir de todos os pares de treinamento, o qual será utilizado na fase de recomendação do UPOD.

Algorithm 1 Fase de Treinamento do UPOD

Require: grafo bipartido usuário-item G , conjunto de atributos de usuários $Feat$, número de grupos k , valores considerados como parâmetro para a difusão Λ , métrica M de avaliação do melhor $\lambda \in \Lambda$ para a difusão híbrida.

```

1:  $CjTreinamento \leftarrow \{\}$ 
2:  $U_{train} \leftarrow usuarios(G, Feat)$ 
3:  $grupos[1..k] \leftarrow kModes(U_{train}, k)$ 
4: for  $i = 1$  to  $i = k$  do
5:    $\lambda_{best}[i] \leftarrow RepeatHybrid(G, grupos[i], \Lambda, M)$ 
6:   for each usuário de  $grupos[i]$  do
7:      $ValorAtributos \leftarrow ExtraiValor(grupos[i].user, Feat)$ 
8:      $CjTreinamento \leftarrow CjTreinamento \cup \{\langle ValorAtributos, \lambda_{best}[i] \rangle\}$ 
9:   end for
10: end for
11:  $classificador \leftarrow TreinaClassificador(CjTreinamento)$ 
12: return  $classificador$ 

```

3.2. Fase de Recomendação

Na fase de recomendação, descrita no algoritmo 2, para cada usuário alvo u_a , são lidos seus atributos (passo 1), os quais são fornecidos como entrada para o classificador treinado na fase de treinamento. O classificador fornece como saída a predição do valor de λ_{u_a} mais apropriado para o referido usuário u_a (passo 2). No passo 3, tendo como entradas o grafo bipartido de usuários-itens G , o usuário alvo u_a , o valor predito λ_{u_a} e o tamanho da lista de recomendações L , UPOD aplica o algoritmo *Hybrid* (eq. 5 e 6) e fornece uma lista de itens recomendados para u_a , ordenada de forma decrescente pelo valor do recurso r'' .

Usando o valor λ_{u_a} predito para o usuário alvo u_a , UPOD fornece uma combinação de itens resultantes de difusões MD e HS, gerando uma lista de

recomendação personalizada para u_a , a qual combina novidade e popularidade dos itens da base de dados.

Algorithm 2 Fase de Recomendação do UPOD

Require: conjunto de atributos de usuários $Feat$, usuário alvo u_a , *classificador*, grafo bipartido usuário-item G , número de itens na lista de recomendação L .

- 1: $ValorAtributos \leftarrow ExtraiAtributos(u_a, Feat)$
 - 2: $\lambda_{u_a} \leftarrow classificador(ValorAtributos)$
 - 3: $listaRecomendacao \leftarrow Hybrid(G, u_a, \lambda_{u_a}, L)$
 - 4: **return** $listaRecomendacao$
-

UPOD foi testado e avaliado segundo as métricas descritas na próxima seção.

4. Métricas de avaliação

Após a geração das listas de itens para cada usuário alvo, é necessário medir o quanto de fato estes itens correspondem ao que o usuário gostaria de receber. Para este propósito, diversas métricas de avaliação das listas de recomendação geradas pelos SR foram utilizadas na literatura, dentre as quais as mais populares são: *Recall*, *Precision*, *Rankscore* e *Novelty* [Zhou et al. 2010, Liu and Liou 2011, Carrer-Neto et al. 2012, Lu et al. 2013, Zhou et al. 2015]. Estas métricas são calculadas em um conjunto de dados à parte daquele usado no treinamento do SR, chamado de conjunto de validação. É importante ressaltar que apenas são considerados usuários do conjunto de validação presentes também no conjunto de treinamento, referenciados como usuários válidos, U_{valid} , para que não se forme vértices isolados em G . A seguir, cada métrica é definida.

4.1. Recall

Recall é utilizada para medir a proporção de itens do conjunto de validação que estão na lista de recomendação, para um determinado usuário alvo:

$$Re_u(L) = \frac{d_u(L)}{I_u} \quad (7)$$

onde $d_u(L)$ é o número de itens relacionados com u no conjunto de validação e que também estão presentes na lista de recomendação L ; I_u é o número total de itens relacionados com u no conjunto de validação.

Este processo normalmente é repetido para cada usuário válido no conjunto de validação, sendo seu valor final calculado como a média dos valores parciais obtidos para cada usuário:

$$\frac{\sum_{u \in U_{valid}} Re_u(L)}{|U_{valid}|}, \quad (8)$$

onde $|U_{valid}|$ representa a quantidade de usuários válidos do conjunto de validação. Quanto maior seu valor, mais itens no conjunto de validação correspondem a itens recomendados.

4.2. Precision

Precision mede a proporção dos itens na lista de recomendação do usuário que também correspondem a ele no conjunto de validação:

$$Pe_u(L) = \frac{d_u(L)}{L} \quad (9)$$

onde $d_u(L)$ é o número de itens relacionados com u no conjunto de validação, presentes na lista de recomendação e L é o tamanho da lista de recomendação. Da mesma forma, calculando para cada usuário presente no conjunto de validação, vem:

$$\frac{\sum_{u \in U_{valid}} Pe_u(L)}{|U_{valid}|}, \quad (10)$$

onde $|U_{valid}|$ representa a quantia de usuários válidos presentes no conjunto de validação. Quanto maior seu valor, mais itens na lista de recomendação correspondem a itens no conjunto de validação.

4.3. RankScore

A métrica *RankScore* é geralmente considerada em trabalhos relacionados a algoritmos baseados em difusão, tais como [Zhou et al. 2010, Zeng et al. 2013, Zhang and Zeng 2015, Zhou et al. 2015]. Nela é considerada a posição de cada item na lista de predição gerada para o usuário alvo, de forma que quanto mais próximo das últimas posições o item estiver (posições com valores de recurso mais significativos na lista de recomendação), menor seu resultado. Obviamente, é levado em conta se um determinado item relacionado ao usuário na lista de predição corresponde ou não a um item relacionado a ele no conjunto de validação:

$$RankScore_u = \sum_{i \in I_u} \frac{rank(i)}{|I_u|} \quad (11)$$

onde I_u é o conjunto de itens relacionados a u no conjunto de validação e $rank(i)$ representa o *ranking* do item i . Se i corresponde a um item relacionado a u no conjunto de validação, $rank(i)$ é dado por:

$$rank(i) = \frac{|predictionList| - pos(i)}{|Unknown_{items}|} \quad (12)$$

caso contrário, é dado por:

$$rank(i) = |predictionList| + term \quad (13)$$

com

$$term = \frac{|Unknown_{items}| - |predictionList| + 1}{2} \quad (14)$$

onde $|predictionList|$ é o tamanho da lista de predição, $|Unknown_{items}|$ é o número de itens com os quais o usuário não havia tido interação inicialmente, e $pos(i)$ é a posição do item i na lista de predição, com $1 \leq pos(i) < |predictionList|$. Tal como ocorre

nas demais métricas, o resultado final é calculado pela média dos valores parciais obtidos para cada usuário do conjunto de validação:

$$\frac{\sum_{u \in U_{valid}} RankScore_u}{|U_{valid}|} \quad (15)$$

onde $|U_{valid}|$ representa a quantidade de usuários válidos presentes no conjunto de validação e $RankScore_u$ é o resultado parcial para cada usuário u . Portanto, quanto menor seu valor, mais itens relacionados aos usuários no conjunto de validação estarão presentes nas primeiras posições das listas de recomendação.

4.4. Novelty

Novelty tem por finalidade medir a capacidade do SR em gerar resultados inesperados, isto é, mede a imprevisibilidade relativa à popularidade global [Zhou et al. 2010]. Logo, dado um item i , a probabilidade de um usuário u aleatoriamente selecioná-lo é dada por:

$$\frac{w(i)}{|U|}, \quad (16)$$

onde $w(i)$ é o grau do nó i no grafo e $|U|$ representa o número de usuários no sistema. Então, sua medida de imprevisibilidade para cada usuário alvo u do conjunto de validação é dado por:

$$Nov_u = \sum_{i \in I_u} \log_2 \frac{|U|}{w(i)}, \quad (17)$$

onde I_u é o conjunto de itens relacionados com o usuário u no conjunto de validação, os quais estão presentes na lista de recomendação. Portanto, assim como nas outras métricas, o resultado final é calculado pela média dos resultados parciais:

$$\frac{\sum_{u \in U_{valid}} Nov_u}{|U_{valid}|}, \quad (18)$$

onde $|U_{valid}|$ representa a quantidade de usuários válidos do conjunto de validação. Quanto maior seu valor, mais itens inesperados (novidades) são recomendados aos usuários.

5. Resultados Experimentais

Os experimentos foram conduzidos em duas bases de dados bem conhecidas na literatura: Movielens e Bookcross. A base de dados Movielens contém 911 usuários, 1.556 itens e 79.324 interações entre ambos. Já na base de dados Bookcross, foi selecionado um subconjunto aleatório para os experimentos com 3.434 usuários, 27.138 itens e 36.191 interações entre ambos.

Para os experimentos foi considerada a divisão randômica do conjunto total de dados, sendo 80% reservado para treinamento e 20% para validação, assim como em [Zeng et al. 2013]. O processo é repetido k vezes, de forma que a cada iteração tem-se uma divisão randômica independente dos conjuntos de treinamento e validação. Ao final das k iterações, calcula-se a média de cada uma das métricas de avaliação.

O espaço de atributos considerado para o agrupamento de usuários através do algoritmo $k - Modes$ na base de dados Movielens foi composto de: idade (foram considerados intervalos de 5 em 5 anos, tais como “5_10”, “10_15”, “15_20”, etc), sexo (m ou f) e localização (no formato “pais_estado”, tal como “US_CA”); já no caso da base de dados Bookcross, os atributos considerados foram: idade (sob o mesmo intervalo de 5 em 5 anos) e localização (nome do país). O classificador adotado foi o SVM [Witten et al. 1999] e a quantia de grupos (k) utilizada foi 5 (este valor foi obtido experimentalmente). Foram consideradas listas de recomendação com $|L| = 30$ posições, para todos experimentos. O algoritmo UPOD foi executado com as mesmas configurações que o FC tradicional [Ricci et al. 2010] e o *Hybrid* [Zhou et al. 2010]. Os resultados comparativos considerando as quatro métricas apresentadas na seção anterior encontram-se na tabela 1 para a base de dados Movielens e na tabela 2 para a base de dados Bookcross.

Tabela 1. Resultados obtidos com a base de dados Movielens.

Algoritmo	RankScore		Recall		Precision		Novelty	
	média	σ	média	σ	média	σ	média	σ
FC	0.493	0.002	0.028	0.002	0.0125	5.2×10^{-4}	3.1	0.2
<i>Hybrid</i>	0.09	0.04	0.25	0.06	0.12	0.06	69.5	0.7
UPOD	0.078	0.001	0.295	0.005	0.135	0.001	85.2	1.4

Tabela 2. Resultados obtidos com a base de dados Bookcross.

Algoritmo	RankScore		Recall		Precision		Novelty	
	média	σ	média	σ	média	σ	média	σ
FC	0.371	0.004	3.6×10^{-4}	1.32	4.2×10^{-5}	1.09	0.04	0.01
<i>Hybrid</i>	0.361	0.099	0.001	0.099	2.3×10^{-4}	0.09	0.26	0.11
UPOD	0.36	0.01	0.002	2.4×10^{-4}	4.3×10^{-4}	3.3×10^{-5}	0.52	0.06

Pode-se observar nas tabelas 1 e 2 que o algoritmo UPOD obteve resultados melhores em todas as métricas, sendo superior tanto à abordagem FC tradicional quanto ao *Hybrid*, nas duas bases de dados testadas. Para avaliar a relevância estatística de cada resultado do algoritmo UPOD em comparação com os demais, foi realizado o Teste t considerando um grau de confiança igual a 95% ($\alpha = 0.05$) e $N = 10$. Neste teste, UPOD demonstrou superioridade estatística em todas as métricas em relação ao FC tradicional, na base de dados Movielens; já na base Bookcross, houve superioridade estatística do UPOD em relação ao FC tradicional nas métricas *RankScore* e *Novelty*. Em relação ao *Hybrid*, UPOD foi superior em *Novelty* nas duas bases de dados e em *Recall* na base Movielens.

Enfim, pela análise dos resultados obtidos e a respectiva relevância estatística, podemos concluir que o algoritmo UPOD obteve resultados significativos, especialmente quando avaliado pelo nível de novidade nas recomendações. Isto indica que, de fato, o aprendizado do grau de combinação (λ) para grupos de usuários com perfis similares influencia positivamente no processo de difusão e, portanto, melhora a qualidade das recomendações geradas.

6. Conclusão

Neste artigo foi apresentado um novo algoritmo, UPOD, com o qual é possível determinar o grau da difusão de itens populares ou novidades, de acordo com o aprendizado do perfil do usuário e gerar uma recomendação personalizada através da combinação apropriada destes tipos de itens. Foi demonstrado, através de alguns experimentos com quatro métricas bastante utilizadas na literatura, que o algoritmo UPOD pode melhorar a qualidade das recomendações quando comparado ao *Hybrid* e também ao algoritmo de filtragem colaborativa tradicional. Foram utilizadas duas bases de dados conhecidas para validar a proposta – Movielens e Bookcross, e os resultados mostraram-se bastante promissores.

Como próximos passos, pode-se citar a definição automática do número de grupos mais apropriado para cada base de dados. Um passo futuro relevante seria também permitir um maior número de difusões no grafo, a exemplo do que é feito no algoritmo SLD [Zeng et al. 2013].

Referências

- Bobadilla, J., Ortega, F., Hernando, a., and Gutiérrez, a. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46:109–132.
- Carrer-Neto, W., Hernández-Alcaraz, M. L., Valencia-García, R., and García-Sánchez, F. (2012). Social knowledge-based recommender system. Application to the movies domain. *Expert Systems with Applications*, 39(12):10990–11000.
- Chen, B.-l., Zeng, A., and Chen, L. (2015). The effect of heterogeneous dynamics of online users on information filtering. *Physics Letters A*, 379(43-44):2839–2844.
- Choi, K., Yoo, D., Kim, G., and Suh, Y. (2012). A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electronic Commerce Research and Applications*, 11(4):309–317.
- Christakopoulou, E. and Karypis, G. (2016). Local Item-Item Models For Top-N Recommendation. *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, pages 67–74.
- Huang, Z. (1998). Extensions to the k -Means Algorithm for Clustering Large Data Sets with Categorical Values. 2(3):283–304.
- Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender Systems: An Introduction*. Cambridge University Press, New York, NY, USA, 1st edition.
- Kotkov, D., Wang, S., and Veijalainen, J. (2016). A survey of serendipity in recommender systems. *Knowledge-Based Systems*, 111:180–192.
- Liu, D.-R. and Liou, C.-H. (2011). Mobile commerce product recommendations based on hybrid multiple channels. *Electronic Commerce Research and Applications*, 10(1):94–104.
- Liu, H., Hu, Z., Mian, A., Tian, H., and Zhu, X. (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, 56:156–166.

- Lu, J., Shambour, Q., Xu, Y., Lin, Q., and Zhang, G. (2013). A web-based personalized business partner recommendation system using fuzzy semantic techniques. *Computational Intelligence*, 29(1):37–69.
- Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. (2015a). Recommender system application developments: A survey. *Decision Support Systems*, 74:12–32.
- Lü, L., Medo, M., Ho, C., Zhang, Y.-c., and Zhang, Z.-k. (2012). Recommender systems. *Physics Reports*, 519(1):1–49.
- Lu, Z., Dou, Z., Lian, J., Xie, X., and Yang, Q. (2015b). Content-based collaborative filtering for news topic recommendation. *AAAI 2015: Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence*, pages 217–223.
- Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. (2010). *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition.
- Sanchez-Moreno, D., Gil Gonzalez, A. B., Munoz Vicente, M. D., Lopez Batista, V. F., and Moreno Garcia, M. N. (2016). A collaborative filtering method for music recommendation using playing coefficients for artists and users. *Expert Systems with Applications*, 66:1339–1351.
- Shambour, Q. and Lu, J. (2015). An effective recommender system by unifying user and item trust information for B2B applications. *Journal of Computer and System Sciences*, 81(7):1110–1126.
- Witten, I. H., Frank, E., Trigg, L., Hall, M., Holmes, G., and Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with Java implementations. In Kasabov, N. and Ko, K., editors, *Proceedings of the ICONIP/ANZIIS/ANNES'99 Workshop on Emerging Knowledge Engineering and Connectionist-Based Information Systems*, pages 192–196. Dunedin, New Zealand.
- Zeng, W., Zeng, A., Shang, M. S., and Zhang, Y. C. (2013). Information filtering in sparse online systems: Recommendation via semi-local diffusion. *PLoS ONE*, 8(11).
- Zhang, F.-G. and Zeng, A. (2015). Information Filtering via Heterogeneous Diffusion in Online Bipartite Networks. *PLoS ONE*, 10(6):e0129459.
- Zhang, Y.-C., Blattner, M., and Yu, Y.-K. (2007). Publisher's Note: Heat Conduction Process on Community Networks as a Recommendation Model [Phys. Rev. Lett. 99, 154301 (2007)]. *Physical Review Letters*, 99(16):169902.
- Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R., and Zhang, Y.-C. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences of the United States of America*, 107(10):4511–4515.
- Zhou, T., Ren, J., Medo, M., and Zhang, Y. C. (2007). Bipartite network projection and personal recommendation. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 76(4):1–7.
- Zhou, Y., Zeng, A., and Wang, W. H. (2015). Temporal effects in trend prediction: Identifying the most popular nodes in the future. *PLoS ONE*, 10(3):1–10.

Um algoritmo GVNS usando o framework KHE para resolver problemas de programação de horários em escolas

Ulisses Rezende Teixeira¹, Marcone Jamilson Freitas Souza², Sérgio Ricardo de Souza¹

¹Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG),
Belo Horizonte, MG, 30510-000, Brasil

²Universidade Federal de Ouro Preto (UFOP),
Ouro Preto, MG, 30510-000, Brazil

ulisses.rezende@gmail.com, marccone@iceb.ufop.br, sergio@dppg.cefetmg.br

Abstract. *This paper presents an algorithm based on General Variable Neighborhood Search (GVNS) for solving High School timetabling problems. The implemented GVNS algorithm uses KHE engine and explores the solution space with four movements: MEET SWAP, MEET TIME CHANGE, MEET BLOCK SWAP and KEMPE TIMES. It was tested in public instances from International Timetabling Competition of 2011 (ITC 2011) and its results were compared with those of the Goal Solver algorithm, winner of the competition. The analysis of results obtained in this work allowed to verify that the method is promising, indicating its strong adherence to this class of problems.*

Resumo. *Este trabalho apresenta um algoritmo baseado no método General Variable Neighborhood Search (GVNS) para resolver problemas de programação de horários em escolas. O algoritmo GVNS usa a biblioteca KHE e explora o espaço de soluções com quatro tipos de movimentos: troca de recursos, troca de bloco de recursos, troca de horários e cadeia Kempe. O algoritmo foi testado em problemas-teste da Competição Internacional de Programação de Horários ocorrida em 2011 e seus resultados foram comparados com aqueles gerados pelo algoritmo Goal Solver, vencedor da competição. A análise dos resultados obtidos com o algoritmo implementado neste trabalho permitiu verificar que o método é promissor, indicando sua forte aderência a esta classe de problemas.*

1. Introdução

Os problemas de programação de horários (*timetabling problems*) têm, como objetivo, distribuir eventos em horários, respeitando restrições previamente determinadas. Uma classe de especial interesse desses problemas, denominados de forma geral como problemas de programação de horários em escolas, são os associados à atribuição de horários, professores, alunos e salas de aula para uma coleção de turmas, de tal forma que nenhum aluno deva comparecer a duas atividades simultaneamente e nenhum professor ministre mais do que uma atividade no mesmo período de tempo. Trata-se, portanto, segundo [Kingston 2010], de uma variação do problema de programação de horários original, que pode, conforme aponta [Schaefer 1999], ser classificado segundo três classes diferentes de problemas, na forma como segue:

- *School Timetabling*: restringe a alocação de professores a horários de uma determinada turma. Usado em especial para escolas de ensino básico (níveis fundamental e médio), em que a matrícula é realizada na forma seriada em blocos de disciplinas de um mesmo período letivo;
- *Course timetabling*: orientado à alocação de disciplinas, sendo cada disciplina parte de um curso e as turmas formadas por alunos de vários cursos. Trata-se de um modelo utilizado tipicamente por instituições de ensino superior, que têm a matrícula na forma de créditos em disciplinas;
- *Examination timetabling*: utilizado no caso de alocação de exames (avaliações) para disciplinas e turmas.

O problema de alocação de horários para escolas de ensino médio é também conhecido como problema de alocação de professores a turmas e consiste em determinar a melhor distribuição de aulas e seus respectivos professores na grade de horários de cada turma, respeitando-se determinadas restrições, como, por exemplo, disponibilidades de horário de cada professor, quantidade de aulas consecutivas, etc. A qualidade da solução é definida de acordo com uma função de avaliação calculada com base na ocorrência ou não de determinadas penalidades, que podem ser fortes, no caso de inviabilizar uma determinada solução, ou fracas, no caso de apenas determinar o grau de aceitação da solução.

Esta foi a categoria de problemas escolhida para o ITC 2011 (*International Timetabling Competition 2011*), terceiro evento internacional desta categoria de problemas, organizado para incentivar estudos e trabalhos no tema.

Todo o interesse demonstrado por esta classe de problemas se baseia em 3 principais pontos, de acordo com [Schaerf 1999]:

- Dificuldade de encontrar uma solução: devido à grande quantidade de restrições, encontrar uma solução viável se torna uma tarefa bastante árdua e que pode levar dias de trabalho manual, de acordo com a quantidade de recursos (turmas, professores, horários) envolvidos.
- Importância prática: ter um quadro de horários é uma necessidade básica de todas as instituições de ensino. Um bom quadro de horários, por outro lado, pode impactar a vida de uma grande quantidade de indivíduos, sejam eles alunos ou professores e pode, inclusive, impactar na eficiência de professores e no desempenho de alunos.
- Importância teórica: é demonstrado, de acordo com [Even et al. 1975], que problemas de programação de horários são da classe NP-difícil, sendo, portanto, objeto de estudo o desenvolvimento de algoritmos eficientes para resolvê-los.

Neste sentido, o presente artigo propõe o estudo da solução do problema de interesse por meio de um algoritmo baseado na metaheurística *General Variable Neighborhood Search* (GVNS) [Hansen et al. 2008]. Esse algoritmo foi testado em problemas-teste disponibilizados no ITC 2011, tendo sido seus resultados comparados com aqueles obtidos pelo algoritmo *Goal Solver* [da Fonseca et al. 2016], que foi o vencedor da competição. Cabe ressaltar que o algoritmo *Goal Solver* utiliza um procedimento híbrido entre as metaheurísticas *Simulated Annealing* (SA) e *Iterated Local Search* (ILS).

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta o formato XHSTT utilizado para a definição dos problemas-teste do ITC 2011. A Seção 3 apresenta a *engine* KHE, utilizada como base para implementação do algoritmo GVNS.

A Seção 4 apresenta o algoritmo implementado, seus conceitos de vizinhança, função de avaliação e alguns detalhes de funcionamento. Na Seção 5 são apresentados os principais resultados obtidos e, finalmente, na Seção 6 são apresentadas as conclusões.

2. Formato XHSTT

O ITC 2011, mais recente evento de competição da área de *timetabling*, focou no modelo de escola de ensino médio (*high school timetabling*), apresentando um grande avanço: a definição de um formato padrão por [Post et al. 2014], que representa diferentes entidades e restrições, agrupados em problemas-teste e organizados em um repositório público. Este formato é conhecido como XHSTT (XML *High School TimeTabling*). Esta representação é organizada em 3 entidades principais: recursos, eventos e restrições. Uma solução para um determinado problema-teste consiste em alocar recursos a eventos, respeitando as restrições definidas no próprio problema-teste. Este mesmo formato permite também modelar a solução encontrada, sendo possível comparar soluções obtidas através de variados métodos e algoritmos. As soluções construídas neste formato podem ser avaliadas através do software HSEval, desenvolvido por [Kingston 2010].

Este novo formato foi escrito no padrão XML (*eXtensible Markup Language*) e pode ser usado livremente, além de estar em constante evolução. Em 2014 eram 21 problemas-teste públicos, usadas em especial para a primeira fase do ITC 2011, evoluindo para 48 problemas-teste diferentes em 2017, construídas por representantes de diferentes países. Esses problemas-teste representam os mais variados cenários, dos mais simples aos mais complexos, contendo, inclusive, cenários reais de escolas de 12 países com seus complicadores reais. A definição desse formato padroniza os estudos e tem, como objetivo, estimular ainda mais o estudo desta categoria de problemas.

Basicamente, o formato define três entidades principais, conforme proposto por [Post et al. 2014]:

- Horários e recursos: um horário consiste em um período de tempo disponível para alocação de um determinado recurso. Já os recursos podem ser categorizados em três subitens: turmas, professores e salas.
- Eventos: um evento representa uma aula, podendo ser associado a um horário e a um recurso.
- Restrições: uma restrição representa uma limitação de vínculo de um recurso ou um horário a um determinado evento. Podem ser restrições básicas de agendamento, restrições de recursos ou restrições de horários. Além disso, cada restrição pode ser definida como forte (*hard*) ou fraca (*soft*), sendo a primeira determinante para garantir soluções viáveis e a segunda responsável pela qualidade da solução determinada pelo valor da função objetivo. Atualmente, o formato oferece 16 tipos de restrições, sendo possível especificar, por exemplo, horários preferidos de professores, disponibilidade de recursos, dentre outros. A Tabela 1 exhibe todas as 16 restrições possíveis para os problemas-teste XHSTT.

Tabela 1. Restrições definidas no formato XHSTT [Post et al. 2014]

Restrições básicas de agendamento		
Item	Tipo de restrição	Descrição
1	Atribuir horário	Atribui um horário a cada evento
2	Atribuir recurso	Atribui recurso a cada evento
3	Definir preferência de horário	Indica que alguns eventos possuem preferência por alguns horários
4	Definir preferência de recurso	Indica que alguns eventos possuem preferência por alguns recursos

Restrições de evento		
Item	Tipo de restrição	Descrição
5	Vincular eventos	Atribui um conjunto de eventos a um mesmo horário de início
6	Espalhar eventos	Distribui eventos de forma uniforme nos horários disponíveis
7	Evitar quebra de atribuições	Para cada evento, atribui um dado recurso a todos os seus encontros
8	Distribuir quebra de eventos	Para cada evento, distribua-o entre um número mínimo e um número máximo de encontros de uma dada duração
9	Quebrar eventos	Impõe limites no número de encontros não-consecutivos criados por um evento e sua duração

Restrições de recursos		
Item	Tipo de restrição	Descrição
10	Evitar conflitos	Evita que se atribua um mesmo recurso a mais de um evento por horário
11	Evitar indisponibilidade	Estabelece que certos recursos são indisponíveis em certos horários
12	Limitar carga de trabalho	Limita a carga total de trabalho de um recurso
13	Limitar horários vagos	O número de horários vagos em cada grupo de horários deve ficar limitado entre um limite mínimo e máximo para cada recurso
14	Limitar horários ocupados	O número de horários ocupados em cada grupo de horários deve ficar limitado entre um limite mínimo e máximo para cada recurso
15	Reduzir horários espalhados de um recurso	Força que a alocação de atividades de um dado recurso seja agrupada em certos grupos de horários

3. Biblioteca KHE

Segundo [Kingston 2012], “*KHE é uma biblioteca de código livre desenvolvida em ANSI C e distribuída sobre a licença pública GNU, que tem como objetivo principal prover métodos rápidos e robustos para a solução dos problemas-teste de timetabling para escolas expressas no formato XHSTT*”.

A biblioteca provê todos os métodos necessários para a manipulação desses problemas-teste, disponibilizando ações para a leitura e escrita dos arquivos XML definidos, assim como para a inclusão e alteração de recursos e alocações destes recursos a eventos, além de manipulação das restrições e tratamento de todas as variáveis disponíveis para a solução dos problemas-teste.

Esta biblioteca provê também, além dos métodos auxiliares já descritos, toda a estrutura de dados para a manipulação dos dados em memória e um algoritmo para a

Algoritmo 1 Algoritmo GVNS

Entrada: Solução inicial s_0 , Total de vizinhanças r' , Total de vizinhanças de refinamento r , número de vizinhos analisados no VND $MaxViz$, Tempo limite de execução $timeLimit$.

Saída: Melhor solução s encontrada.

```

1:  $s \leftarrow s_0$ 
2: enquanto tempoDecorrido <  $timeLimit$  faça
3:    $k \leftarrow 1$ ;
4:   enquanto  $k \leq r'$  faça
5:      $s' \leftarrow Shake(s, k)$ ;
6:      $s'' = VND(s', r, MaxViz)$ ;
7:     se  $f(s'') < f(s)$  então
8:        $s \leftarrow s''$ ;
9:        $k \leftarrow 1$ ;
10:    senão
11:       $k \leftarrow k + 1$ ;
12:    fim se
13:  fim enquanto
14: fim enquanto
15: return  $S$ ;

```

construção de uma solução de forma rápida, chamado *KheGeneralSolve* [Kingston 2014]. Esta rotina garante a geração de uma solução para qualquer problema-teste, apesar de não garantir a viabilidade ou a qualidade da solução encontrada.

4. Algoritmo GVNS proposto

Para resolver o problema de programação de horários em escolas, no presente artigo é apresentado um algoritmo baseado na metaheurística *General Variable Neighborhood Search* – GVNS [Hansen et al. 2008]. Esta metaheurística explora o espaço de soluções por meio de trocas sistemáticas de vizinhanças, previamente ordenadas, e tem, como procedimento de busca local, o algoritmo *Variable Neighborhood Descent* – VND [Mladenović and Hansen 1997]. O pseudocódigo do algoritmo GVNS implementado está apresentado no Algoritmo 1.

No algoritmo GVNS, parte-se de uma solução inicial s_0 (linha 1 do Algoritmo 1). Enquanto o critério de parada não for atendido, ou seja, enquanto houver tempo disponível (linha 2), o método passa por um processo iterativo (linhas 2 a 14), começando a exploração do espaço de soluções a partir da primeira vizinhança (linha 3). Em seguida, é gerado nessa vizinhança um vizinho s' qualquer da solução corrente s (função *Shake* na linha 5) e, sobre ele, é aplicada uma busca local (linha 6, cujo procedimento é descrito pelo Algoritmo 2 na subseção 4.1). Se a solução s'' proveniente desse refinamento for melhor que a solução corrente s , ela passa a ser a nova solução corrente e retorna-se à primeira vizinhança; caso contrário, passa-se para a próxima vizinhança (linha 11). Se todas as r' vizinhanças forem exploradas e ainda houver tempo, reinicia-se a busca a partir da primeira vizinhança.

As subseções seguintes detalham o procedimento VND do algoritmo GVNS, bem

Algoritmo 2 Algoritmo VND

Entrada: Solução inicial s_0 , Total de vizinhanças do refinamento r , Número máximo de vizinhos em cada vizinhança $MaxViz$.

Saída: Melhor solução s encontrada.

```

1:  $k \leftarrow 1$ ;
2:  $s \leftarrow s_0$ ;
3: enquanto  $k \leq r$  faça
4:    $iter \leftarrow 1$ ;
5:   melhorou  $\leftarrow$  falso;
6:   enquanto existir vizinho e  $iter \leq MaxViz$  faça
7:      $iter \leftarrow iter+1$ ;
8:      $s' \leftarrow$  próximo vizinho aleatório de  $s$  na vizinhança  $k$ ;
9:     se  $f(s') < f(s)$  então
10:       $s \leftarrow s'$ ;
11:      melhorou  $\leftarrow$  verdadeiro;
12:   fim se
13: fim enquanto
14: se melhorou então
15:    $k \leftarrow 1$ ;
16: senão
17:    $k \leftarrow k + 1$ ;
18: fim se
19: fim enquanto
20: return  $S$ ;

```

como as vizinhanças usadas e a função de avaliação das soluções.

4.1. VND

O procedimento VND (*Variable Neighborhood Descent*), acionado na linha 6 do Algoritmo 1, é um método de refinamento que utiliza também trocas sistemáticas de vizinhanças, previamente ordenadas, para explorar o espaço de soluções do problema. No procedimento implementado, cujo pseudocódigo está apresentado pelo Algoritmo 2, tem-se, como dados de entrada, uma solução inicial s_0 , as r vizinhanças previamente ordenadas e o número de vizinhos $MaxViz$ analisados por vizinhança.

O Algoritmo 2 começa o refinamento a partir da primeira vizinhança (linha 1) aplicada sobre a solução corrente (linha 2). Enquanto não forem analisadas todas as k'_{max} vizinhanças, passa-se por um processo iterativo (linhas 3 a 13), em que, a cada iteração, busca-se o primeiro vizinho em no máximo de $maxviz$ que seja de melhora. Se esse vizinho escolhido tiver avaliação melhor que a da solução corrente, ele passa a ser a nova solução corrente (linha 10) e retorna-se à primeira vizinhança (linha 15); caso contrário, passa-se para a próxima vizinhança (linha 17). O procedimento é encerrado quando não for encontrado vizinhos de melhora em relação à solução corrente em nenhuma das r vizinhanças.

4.2. Vizinhanças

Na implementação do algoritmo GVNS foram utilizadas $r' = 10$ vizinhanças, cada qual definida pela aplicação sucessiva do movimento KEMPE descrito na subseção 4.2.4. Na primeira vizinhança o movimento é aplicado uma vez, na segunda vizinhança duas vezes e assim por diante até o limite de 10 aplicações deste movimento na vizinhança de número 10.

Já na implementação do algoritmo VND foram utilizadas $r = 3$ vizinhanças, cada qual baseada nos seguintes movimentos, nesta ordem:

- Troca de recursos: consiste na aplicação do movimento MEET SWAP, descrito na subseção 4.2.1.
- Troca de horário: consiste na aplicação do movimento MEET TIME CHANGE, descrito na subseção 4.2.3.
- Troca de bloco de recursos: consiste na aplicação do movimento MEET BLOCK SWAP, descrito na subseção 4.2.2.

As subseções a seguir descrevem cada um dos movimentos utilizados na exploração do espaço de soluções do problema, todos eles já descritos na literatura especializada.

4.2.1. Troca de recursos (MEET SWAP)

Este movimento identifica dois horários e realiza a troca dos recursos alocados a eles.

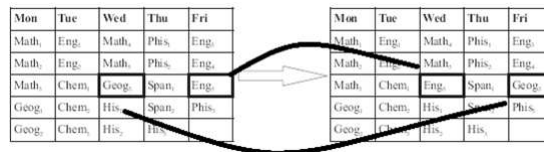


Figura 1. Movimento de troca de recursos.

4.2.2. Troca de bloco de recursos (MEET BLOCK SWAP)

Este movimento é similar ao de troca de recursos; entretanto, se houver recursos adjacentes, os recursos são movidos, mantendo a adjacência, sem perder a contiguidade.

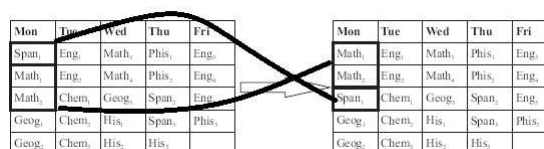


Figura 2. Movimento de troca de recursos em bloco.

4.2.3. Troca de horário (MEET TIME CHANGE)

Neste movimento é selecionado um recurso. Então as associações deste recurso são re-arranjadas para a melhor posição. Todas as possibilidades de rearranjo são analisadas e é escolhida a melhor delas.

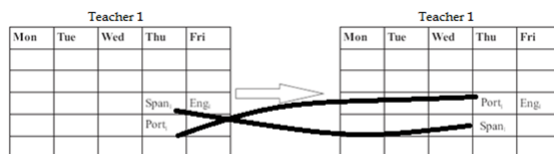


Figura 3. Movimento de troca de horário.

4.2.4. Kempe (KEMPE TIMES)

Este movimento é baseado no algoritmo *Kempe Chain Interchanges* (KCI), proposto por [Johnson et al. 1991] como solução para o problema de coloração de grafos. O movimento foca na solução de conflitos, realizando alguns movimentos simultâneos envolvendo um recurso previamente selecionado no sentido de produzir soluções melhores. O algoritmo implementado, diferentemente do algoritmo KCI original, pode gerar soluções ineficazes, mas a cada execução ele busca a melhor dentre todas as soluções possíveis. No exemplo exibido na Figura 4.2.4, tenta-se resolver o conflito existente mostrado pela janela de horários de um determinado professor.

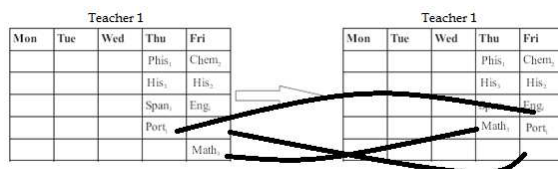


Figura 4. Movimento de cadeia de Kempes.

4.3. Função de avaliação

A função de avaliação do algoritmo, implementada pela biblioteca KHE, retorna um valor que define a quantidade de restrições *hard* e *soft* que não foram atendidas, medindo assim a viabilidade e a qualidade da solução gerada. O retorno da função é um número decimal, sendo a parte inteira o número de restrições *hard* e a parte decimal o número de restrições *soft* não atendidas. Neste artigo, para fins de comparação, os resultados são apresentados como um par xy sendo que x representa o número de restrições *hard* e y o número de restrições *soft*.

5. Resultados

O algoritmo GVNS foi implementado na linguagem C++ usando a IDE Code:Blocks e testado em um notebook Intel Core i5, com 4 GB de RAM sob o sistema operacional Windows 10. Para testar o algoritmo GVNS foram usados 21 problemas-teste disponibilizados na ITC 2011, descritos na Tabela 2. Nesta Tabela, a primeira coluna indica o

problema-teste; a segunda coluna, o número de horários semanais; a terceira coluna representa o número de professores; a quarta coluna, o número de salas; a quinta coluna, o número de turmas e, na última coluna, o número de aulas a serem distribuídas entre os professores.

Tabela 2. Problemas-teste da ITC 2011

Problema-teste	Horários	Profs	Salas	Turmas	Aulas
AustraliaBGHS98	40	56	45	30	1564
AustraliaSAHS96	60	43	36	20	1876
AustraliaTES99	30	37	26	13	806
BrazilInstance1	25	8	-	3	75
BrazilInstance4	25	23	-	12	300
BrazilInstance5	25	31	-	13	325
BrazilInstance6	25	30	-	14	350
BrazilInstance7	25	33	-	20	500
EnglandStPaul	27	68	67	67	1227
FinlandArtificialSchool	20	22	12	13	200
FinlandCollege	40	46	34	31	854
FinlandHighSchool	35	18	13	10	297
SecondarySchool	35	25	25	14	306
GreeceHighSchool1	35	29	-	66	372
GreecePatras3rdHS2010	35	29	-	84	340
GreecePreveza3rdHS2008	35	29	-	68	340
ItalyInstance1	36	29	-	3	133
NetherlandsGEPRO	44	132	80	44	2675
NetherlandsKottenpark2003	38	75	41	18	1203
NetherlandsKottenpark2005	37	78	42	26	1272
SouthAfricaLewitt2009	148	19	2	16	838

O algoritmo GVNS foi executado com os seguintes valores para seus parâmetros:

- *MaxViz*: parâmetro que define a quantidade máxima de vizinhos analisados em cada vizinhança no método VND, definido com o valor de 1.000.000
- *timeLimit*: parâmetro que define o tempo máximo de execução. Foi definido com o mesmo valor utilizado pelo algoritmo GOAL Solver no ITC 2011 em sua primeira fase, ou seja, 1.000 segundos, conforme [da Fonseca et al. 2016].

As soluções iniciais para o algoritmo GVNS foram obtidas através de duas formas:

- Fornecidas pela organização do evento ITC 2011. Uma solução inicial para cada problema-teste.
- Através da execução da função de geração de solução inicial *KHEGeneralSolve* disponível na biblioteca KHE e mencionada anteriormente. Essas soluções foram utilizadas em três dos problemas-teste: AustraliaBGS98, AustraliaSAHA96 e AustraliaTES99. Isso porque a solução inicial calculada apresentou valores melhores do que as divulgadas pela organização do evento.

As soluções iniciais do algoritmo GVNS para os problemas-teste foram as mesmas usadas como solução inicial na primeira etapa da competição, como descrito por

Tabela 3. Comparação de resultados entre os algoritmos GVNS e GOAL Solver

Problema-teste	ITC 2011	Goal Solver (SA + ILS)	GVNS	gap (%)
AustraliaBGHS98	6/450	6/450	4/370	-33,33
AustraliaSAHS96	17/55	14/50	12/51	-14,29
AustraliaTES99	7/163	7/161	7/151	-6,21
BrazilInstance1	0/24	0/12	0/11	-8,33
BrazilInstance4	0/112	0/91	0/94	3,30
BrazilInstance5	0/225	0/164	0/155	-5,49
BrazilInstance6	0/209	0/149	0/148	-0,67
BrazilInstance7	0/330	0/264	0/249	-5,68
EnglandStPaul	0/18.444	0/18.092	0/12542	-30,68
FinlandHighSchool	0/1	0/1	0/1	0,00
FinlandSecondarySchool	0/106	0/86	0/87	1,16
ItalyInstance1	0/28	0/19	0/18	-5,26
NetherlandsGEPRO	1/566	1/566	1/434	-23,32
NetherlandsKottenpark2003	0/1410	0/1409	0/1216	-13,70
NetherlandsKottenpark2005	0/1078	0/1078	0/881	-18,27
SouthAfricaLewitt2009	0/58	0/22	0/24	9,90

[da Fonseca et al. 2016]. Utilizar as mesmas soluções iniciais apresenta vantagens por permitir comparar a qualidade das soluções encontradas pelo algoritmo GVNS e pelo algoritmo vencedor da competição, este último baseado na combinação das metaheurísticas *Simulated Annealing* e ILS.

Na Tabela 3 são apresentados os resultados encontrados pelo algoritmo GVNS (quarta coluna), comparando-os com a solução inicial provida pelo ITC 2011 (segunda coluna) e com os resultados do algoritmo vencedor da competição, o Goal Solver (terceira coluna). Esses resultados do Goal Solver foram obtidos pela sua execução na mesma máquina em que o GVNS foi testado. A coluna *gap* é calculada pela expressão: $gap = f^*(GVNS) - f^*(Goal) / f^*(Goal)$, sendo $f^*(GVNS)$ o resultado alcançado pelo algoritmo GVNS no respectivo problema-teste e $f^*(Goal)$ o alcançado pelo Goal Solver. Quando os valores de avaliação das restrições fortes são diferentes, o *gap* é calculado apenas com base nesses valores. Nos demais casos são usados somente os valores das restrições fracas. Resultados negativos de *gap* indicam que o algoritmo GVNS foi melhor que o algoritmo Goal Solver. Como em cinco dos vinte e um problemas-teste a solução inicial tinha função de avaliação nula tanto em relação às restrições *hard*, quanto em relação às restrições *soft*, isto é, a solução inicial era ótima, esses resultados não foram apresentados nesta Tabela.

Como pode ser observado na Tabela 3, dos dezesseis problemas-teste em que o algoritmo GVNS foi testado, em doze ele ganhou do Goal Solver, em um ele empatou e em três ele perdeu.

6. Conclusões

Este trabalho teve seu foco no problema de programação de horários em escolas. Para resolvê-lo foi implementado um algoritmo baseado na metaheurística GVNS. O algoritmo explora o espaço de soluções com quatro tipos diferentes de movimentos: MEET SWAP, MEET TIME CHANGE, MEET BLOCK SWAP e KEMPE TIMES, sendo os dois primeiros usados também como método de refinamento do VND.

Para testá-lo, foram usados os problemas-teste da ITC 2011 e as mesmas condições dessa competição, que teve o algoritmo Goal Solver como vencedor.

Verificou-se que o algoritmo GVNS foi capaz de melhorar a solução inicial fornecida pela organização da competição e superar o algoritmo vencedor da competição, o Goal Solver, em doze dos dezesseis problemas-teste.

O algoritmo GVNS implementado apresenta como vantagem o fato de ter poucos parâmetros, no caso, a ordem das vizinhanças e os dois critérios de parada: número de vizinhos analisados durante o refinamento pelo VND e o tempo total de execução. Essa é uma vantagem sobre o algoritmo Goal Solver, que tem oito parâmetros, sendo quatro de cada um dos dois métodos usados na hibridização.

Esses resultados mostram a relevância dos métodos com abordagem de busca local, já demonstrado inclusive pelo próprio algoritmo vencedor da competição com o seu algoritmo híbrido, que combina dois métodos de busca local em sua formulação.

Como trabalhos futuros espera-se testar a adição de outras vizinhanças ao algoritmo GVNS.

Agradecimentos

Os autores agradecem ao CNPq, FAPEMIG, CEFET-MG e UFOP pelo apoio ao desenvolvimento deste trabalho. Agradecem, também, aos autores do Goal Solver pela disponibilização do código do algoritmo.

Referências

- [da Fonseca et al. 2016] da Fonseca, G. H. G., Santos, H. G., Toffolo, T. A. M., Brito, S. S., and Souza, M. J. F. (2016). GOAL solver: a hybrid local search based solver for high school timetabling. *Annals of Operations Research*, 239(1):77–97.
- [Even et al. 1975] Even, S., Itai, A., and Shamir, A. (1975). On the complexity of time table and multi-commodity flow problems. In *16th Annual Symposium on Foundations of Computer Science*, pages 184–193. IEEE.
- [Hansen et al. 2008] Hansen, P., Mladenović, N., and Perez, J. A. M. (2008). Variable neighbourhood search: methods and applications. *4OR*, 6:319–360.
- [Johnson et al. 1991] Johnson, D. S., Aragon, C. R., McGeoch, L. A., and Schevon, C. (1991). Optimization by simulated annealing: an experimental evaluation; part ii, graph coloring and number partitioning. *Operations research*, 39(3):378–406.
- [Kingston 2014] Kingston, J. (2014). Khe14: An algorithm for high school timetabling. In *Proceedings of the 10th International Conference of the Practice and Theory of Automated Timetabling*, pages 498–501, York, England.

- [Kingston 2010] Kingston, J. H. (2010). The hseval high school timetable evaluator. <http://www.it.usyd.edu.au/~jeff/hseval.cgi>.
- [Kingston 2012] Kingston, J. H. (2012). A software library for school timetabling. <http://sydney.edu.au/engineering/it/~jeff/khe/>.
- [Mladenović and Hansen 1997] Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100.
- [Post et al. 2014] Post, G., Kingston, J. H., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngas, J., Nurmi, C., Musliu, N., Pillay, N., Santos, H., et al. (2014). Xhstt: an xml archive for high school timetabling problems in different countries. *Annals of Operations Research*, 218(1):295–301.
- [Schaerf 1999] Schaerf, A. (1999). A survey of automated timetabling. *Artificial intelligence review*, 13(2):87–127.

Classificação de Emoções Discretas em Sinais de EEG Utilizando Técnicas de Aprendizado de Máquina

Vancley Oliveira Simão¹, Heitor Scalco Neto¹, Wilian Soares Lacerda¹

¹LABSINE - Laboratório de Sistemas Inteligentes e Embarcados
Departamento de Ciência da Computação – Universidade Federal de Lavras (UFLA)
Caixa Postal: 3037 – CEP: 37.200-000 – Lavras – MG – Brasil

Abstract. *Classify emotions is not a simple task as there is no concrete emotion definition that the researchers consider unanimous. In this paper, is verified the possibility of using Artificial Neural Networks (ANN) and the Random Forest models to classify emotions on EEG signals. The DEAP database was used where it was extracted the Hjorth coefficients: activity, complexity and mobility. After the features were extracted, the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm was applied to select the best features and reduce the dataset dimensionality. For last, the Random Forest and Artificial Neural Networks models were implemented to classify each emotion represented in the PAD emotional state model. After the tests was observed an average accuracy of 87% for the Random Forest model and 64% for the Neural Network model.*

Resumo. *Classificar emoções é uma tarefa complexa. Neste artigo é investigada a possibilidade da utilização de técnicas de aprendizado de máquina para classificar emoções em sinais de EEG. Para o desenvolvimento do projeto foi utilizada a base de dados DEAP em que foram extraídos os coeficientes de Hjorth: atividade, complexidade e mobilidade. Após a extração dos atributos, foi utilizado a técnica t-SNE, para realizar a seleção dos melhores atributos e redução do alto número de dimensões. Após selecionados os melhores atributos, foram implementados um modelo utilizando o algoritmo de Florestas Aleatórias e outro modelo utilizando Redes Neurais Artificiais de Múltiplas Camadas para realizar a classificação de cada uma das emoções representadas no modelo emocional PAD. Ao fim dos testes, foi obtida um taxa de acurácia média de 87% para o modelo utilizando Florestas Aleatórias e 64% para modelo utilizando Redes Neurais de Múltiplas Camadas.*

1. Introdução

O que é uma emoção? Essa é uma das perguntas mais antigas feitas pelo homem, e remete a pensadores antigos como Platão. O estudo das emoções se vê presente em várias áreas do conhecimento, entretanto, defini-la não é uma tarefa fácil o que faz com que pesquisadores não cheguem em uma definição única e absoluta [Drever 1932]. O que se tem certeza é de suma importância do estudo e entendimento das emoções, e do seu papel fundamental em modelos científicos da mente e do comportamento humano [Lerner and Keltner 2000].

Ações, tomada de decisões e até mesmo o pensamento são fortemente influenciados pelo estado emocional de uma pessoa. Emoções são vistas como causas ou efeitos

de outros processos psicológicos e cognitivos do ser humano, como a percepção do que se encontra ao seu redor, atenção e memória [Barrett 2006].

Vários métodos para a classificação de emoções foram propostos utilizando análises de sinais fisiológicos e métricas psicológicas, como a análise de expressões faciais, fala, respiração, dentre outros [Zhang et al. 2017]. Entretanto, a utilização de sinais de EEG (Eletroencefalograma) pode vir a apresentar melhores resultados em vista da sua alta resolução temporal [Sammler et al. 2007].

Neste trabalho, é investigada, através da análise de métricas de acurácia, a utilização dos algoritmos de aprendizado de máquina RNA e Florestas Aleatórias para classificar emoções em sinais de EEG. Foi utilizado a base de dados DEAP [Koelstra et al. 2012] para realizar o treinamento e teste dos classificadores. Por fim, foi feita uma análise comparativa dos dois modelos propostos.

Este artigo está organizado da seguinte forma: na Seção 2 é apresentado brevemente uma revisão da literatura sobre Emoções, dos algoritmos SMOTE e t-SNE, e das técnicas de aprendizado de máquina RNA e Florestas Aleatórias. Na Seção 3 é apresentada a metodologia utilizada no desenvolvimento deste artigo, bem como os materiais utilizados. Por fim, nas sessões 4 e 5 são apresentados os resultados e conclusões obtidas.

2. Revisão Bibliográfica

Esta seção apresenta uma breve revisão bibliográfica sobre Emoções, algoritmo de balanceamento de dados SMOTE, algoritmo de seleção de atributos e redução do número de dimensões t-SNE e, por fim, os algoritmos de aprendizado de máquina Redes Neurais Artificiais (RNA) de Múltiplas Camadas e Florestas Aleatórias.

2.1. Psicologia das Emoções

Vários modelos foram teorizados para caracterizar cada emoção, além de tentar entender a correlação entre as respostas emocionais a determinados estímulos, em conjunto com o estudo das respostas fisiológicas do cérebro. Cientistas influenciados por Darwin e pelos recentes estudos na área da neurociência, acreditam que as emoções mais básicas são fenômenos neurofisiológicos moldados pela seleção natural ao longo do tempo. Assim, o conjunto de padrões de respostas aos estímulos e cenários aos quais os indivíduos foram submetidos, evoluiu juntamente com os humanos [Izard 2007].

Uma emoção básica pode ser considerada como um conjunto de componentes neurais, comportamentais e emocionais gerados de forma natural, automática, e de forma não consciente como resposta a um estímulo [Izard 2007].

Emoções podem ser representadas utilizando utilizando uma perspectiva dimensional [Mauss and Robinson 2009]. Em [Russell 1980] é proposto um modelo bidimensional para se representar emoções. Neste modelo, a primeira dimensão representa o valor de *Valence* e a segunda dimensão representa o valor de *Arousal*. Entretanto, utilizar somente duas dimensões não é o suficiente para representar todas as emoções, o que acarreta em emoções distintas serem representadas no subespaço bidimensional. Assim, no intuito de apresentar uma melhor representação dimensional das emoções, em [Mehrabian 1995a] e [Mehrabian 1996] é apresentado um modelo tridimensional. Neste novo modelo, o valor de *Dominance* é utilizado como uma terceira dimensão.

2.2. SMOTE

Segundo [He and Garcia 2009], podemos definir como dados não-balanceados qualquer conjunto de dados em que a distribuição de amostras para cada classe presente não é igual. Dados não balanceados podem influenciar negativamente a acurácia de um modelo de classificação.

Em [Pereira and Gomes 2016] é verificado a importância do processo de balanceamento de dados em modelos para classificação de emoções em sinais de EEG, no qual são apresentados resultados que reforçam a utilização deste tipo de abordagem para melhorar os modelos de classificação em dados de EEG. O não balanceamento dos dados pode vir a acarretar problemas para técnicas de aprendizado de máquina lineares e não-lineares, como Florestas Aleatórias e RNA. Em vista que essas técnicas apresentam uma tendência de classificar os dados para a classe majoritária, uma vez que levam em conta o valor do erro para o ajuste do modelo de classificação. Para solucionar esse problema, várias técnicas foram desenvolvidas baseadas na manipulação de amostragens criadas a partir da base de dados originais.

Uma forma de se balancear bases de dados está na criação de amostras sintéticas. Dentre as técnicas mais utilizadas para este fim está o algoritmo *Synthetic Minority Over-sampling Technique* (SMOTE) [Chawla et al. 2002]. Neste algoritmo são criadas amostras sintéticas ou artificiais baseadas nas amostras reais que formam a base de dados original.

Para um subconjunto S_{min} que correspondem ao conjunto de amostras que compõem a classe minoritária, são consideradas as K amostras que estão mais próximas a uma amostra $x_i \in S_{min}$. Essas amostras são calculadas de acordo com o algoritmo *K-nearest Neighbors* (KNN) [Altman 1992]. Assim, para se criar uma nova amostra sintética, basta multiplicar uma dessas amostras selecionadas pelo KNN a um número aleatório entre 0 e 1. Por utilizar o algoritmo KNN para a criação das novas amostras a serem adicionadas, deve-se definir o parâmetro K , ou seja, definir o número de amostras que o algoritmo irá levar em consideração no cálculo das novas amostras.

Esta técnica também pode apresentar problemas para os classificadores por não levar em consideração as amostras próximas que pertençam a outras classes. Esta não consideração pode vir a gerar uma sobreposição de classes, o que pode tornar o problema de classificação mais complexo.

2.3. t-Distributed Stochastic Neighbor Embedding

A visualização de conjunto de dados de alta dimensionalidade é um importante problema em diferentes áreas do conhecimento, e influenciado pela necessidade de algoritmos que solucionassem este problema, Van der Maaten em [Van Der Maaten and Hinton 2008] propôs o algoritmo *t-Distributed Stochastic Neighbor Embedding* (t-SNE).

O algoritmo t-SNE é uma extensão do algoritmo *Stochastic Neighbor Embedding* (SNE) proposto em [Hinton and Roweis 2002], como uma tentativa de melhorar o desempenho em vista que o SNE apresenta uma função de custo de difícil de ser otimizada.

O t-SNE minimiza a diferença entre duas distribuições. A primeira distribuição é formada pela similaridade entre pares de amostras do conjunto de dados original. A segunda distribuição é composta pela similaridade entre os pontos do con-

junto de dados original mapeados em um novo espaço de baixa dimensionalidade [Van Der Maaten and Hinton 2012].

Conforme [Van Der Maaten 2014], dado um conjunto de entrada de alta dimensionalidade $X = x_1, x_2, \dots, x_n$ e uma função $d(x_j, x_i)$ que calcula a distância Euclidiana entre dois pontos:

$$d(x_j, x_i) = \|x_j - x_i\|. \quad (1)$$

O objetivo do algoritmo é aprender uma representação de s dimensões onde cada ponto do conjunto original é representado no conjunto $\xi = y_1, y_2, \dots, y_n$ com $y_i \in \mathbb{R}^s$. Em seguida o algoritmo calcula a probabilidade p_{ji} que representa a probabilidade de dois pontos x_i e x_j serem semelhantes:

$$p_{j|i} = \frac{\exp(-d(x_j, x_i)^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-d(x_k, x_i)^2/2\sigma_i^2)}, \quad (2)$$

$$p_{i|i} = 0, \quad (3)$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}. \quad (4)$$

O valor do *kernel* gaussiano σ_i ótimo varia de ponto para ponto, uma vez que para uma distribuição p_i o valor da sua perplexidade condicional u é predefinida. Assim, para encontrar o valor ótimo de σ pode ser utilizada uma simples busca binária. No novo espaço dimensional ξ a similaridade entre dois pontos y_i e y_j , que são representações dos pontos x_i e x_j em um espaço de baixa dimensionalidade (s dimensões), é calculada por:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}. \quad (5)$$

Por fim, acontece a minimização da divergência de Kullback-Leibler entre as distribuições de similaridade peq . Assim, a localização final dos pontos transformados no novo espaço de baixa dimensionalidade pode ser descrita como:

$$C(\xi) = KL(p||q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \quad (6)$$

2.4. Redes Neurais Artificiais

Redes Neurais Artificiais (RNA) são algoritmos inspirados no funcionamento do sistema nervoso humano. Em paralelo ao sistema nervoso humano, uma RNA apresenta neurônios artificiais que realizam sinapses entre si, a fim de trocarem informações. O conjunto desses neurônios e suas sinapses formam uma RNA.

Podemos representar matematicamente um neurônio artificial como:

$$y_k = \sum_{j=1}^m w_{kj} x_j + b_k, \quad (7)$$

onde y_k representa o sinal de saída do neurônio, w_{kj} representa o vetor de pesos sinápticos atribuídos aos neurônios para uma entrada x_j , e b_k representa o valor do bias.

A forma com que os neurônios são conectados entre si em uma RNA, definem estritamente como deverão ser implementados os algoritmos de aprendizado de máquina utilizados nessa rede. Em geral podemos classificar uma RNA, quanto a sua arquitetura, em três tipos: Redes *Feedforward* de camada única, Redes *Feedforward* de múltiplas camadas e Redes Recorrentes.

Nas Redes *Feedforward* de camada única, como sugere o nome, a RNA possui apenas uma camada, assim os neurônios de entrada são conectados diretamente nos neurônios de saída. O fluxo de informações ou *feedback* desse tipo de arquitetura é somente para frente, por isso o nome de *feedforward*. Já nas Redes *Feedforward* de múltiplas camadas um novo tipo de camada intermediária entre os neurônios de entrada e os neurônios de saída é apresentado, chamado de *hidden layer* ou camada oculta. Assim, ao se adicionar essas camadas intermediárias torna-se possível resolver problemas de maior complexidade. Os neurônios de entrada se conectam aos neurônios das camadas ocultas, até se conectarem aos neurônios da camada de saída.

Por fim, nas Redes Recorrentes passa a existir um novo tipo de fluxo de informações na rede, chamado de *feedback loop*. Assim, tanto em uma RNA de camada única ou com múltiplas camadas podem existir neurônios que se conectam com neurônios de uma camada anterior.

As RNA ainda podem ser classificadas de acordo com o tipo do seu algoritmo de aprendizado: aprendizado supervisionado e aprendizado não-supervisionado. No aprendizado supervisionado existe a figura do professor que funciona como um moderador para os questionamentos feitos ao neurônio. Assim, através da resposta correta da rede apresentada pelo professor em conjunto com a resposta apresentada pela rede, o algoritmo consegue calcular um valor de ajuste para os parâmetros da rede em função do valor obtido pelo valor desejado. Já no aprendizado não-supervisionado não existe um professor, assim o aprendizado é feito somente através da análise das características estatísticas dos dados.

2.4.1. Algoritmo *Backpropagation*

Entre os principais algoritmo de aprendizado supervisionado aplicado a RNA está o algoritmo de retropropagação do erro ou *error backpropagation*, que é baseado na regra de aprendizagem por correção de erro, conhecida como regra delta.

Esse algoritmo é formado, basicamente, por um avanço ou propagação e um retrocesso ou retropropagação, onde nesse avanço um sinal aplicado na entrada é processado e propagado para as camadas seguintes mantendo um valor fixo para os pesos sinápticos da rede, gerando uma saída como resposta da rede.

Já no processo de retropropagação, os pesos sinápticos da rede são reajustados de acordo com uma regra de correção de erro, obtido pela subtração da resposta real da rede pela resposta desejada. A partir desse cálculo, esse sinal de erro é propagado para as camadas de trás. Esse processo de ajuste dos pesos sinápticos é denominado mecanismo de controle, e seu principal objetivo é aproximar a resposta atual da rede $y_k(n)$ da resposta

desejada $d_k(n)$. Essa aproximação é obtida através da redução do índice de desempenho $\psi(n)$, [Haykin 2008] que é definido por:

$$\xi(n) = \frac{1}{2}e_k^2(n). \quad (8)$$

Esses ajustes dos pesos sinápticos continuam a acontecer, até que a rede chegue a um estado estável e a resposta seja a desejada. O processo de redução do índice de desempenho descrito acima, $\xi(n)$ compõe a regra de aprendizado denominada regra delta [Widrow and Hoff 1960]. De acordo com a regra delta, o ajuste $\Delta w_{kj}(n)$ aplicado ao peso sináptico w_{kj} no tempo n é definido por:

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n), \quad (9)$$

onde η determina a taxa de aprendizado, v_k se refere ao potencial de ativação do neurônio k , w_{kj} ao peso sináptico da sinapse j pertencente ao neurônio k . Uma vez que a delta pressupõe que o sinal de erro seja mensurável, é necessário que a resposta desejada seja previamente informada por uma fonte externa e que seja acessível o neurônio k . [Haykin 2008]

O ajuste dos pesos a partir da regra delta é denominado Regra Delta Generalizada e é obtido de acordo com a seguinte equação:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha[w_{ji}^{(l)}(n-1)] + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n). \quad (10)$$

2.5. Florestas Aleatórias

Com o desenvolvimento de técnicas de *ensemble* de árvores e o processo de combinar suas respostas de modo que eles votem na classe mais popular, observou-se uma melhora na acurácia de modelos de classificação [Breiman 2001].

O algoritmo de Florestas Aleatórias tem sido bastante utilizado em vista da sua alta capacidade de generalização, além de ser utilizado em tarefas de classificação e regressão. Em [Kotti and Stylianou 2017], o algoritmo é utilizado para o reconhecimento de emoções em trechos de áudio retirados de filmes. Já em [Zhang et al. 2016] são classificadas emoções em músicas, também utilizando o algoritmo de Florestas Aleatórias.

De modo geral, o algoritmo de Florestas Aleatórias ou *Random Forest* (RF) é uma combinação de árvores de decisão, onde cada árvore gerada depende de um subconjunto de dados amostrados de forma independente e que apresentem a mesma distribuição para todas as árvores que formam a floresta.

Podemos definir o algoritmo como um classificador formado por árvores de decisão $h(x, \Theta_k)$, $k = 1$ onde Θ_k são vetores aleatórios independentes e identicamente distribuídos e cada árvore vota a classe que a entrada x é pertencente [Breiman 2001].

Dado um conjunto de classificadores $h_1(x), h_2(x), \dots, h_k(x)$, e como conjunto de treinamento X selecionado aleatoriamente da distribuição de dados Y , segundo [Nugrahaeni and Mutijarsa 2017] podemos definir a função de margem como:

$$margem(X, Y) = media_k I(h_k(X) = Y) - max_{j \neq Y} media_k I(h_k(X) = j), \quad (11)$$

onde a função $I(\cdot)$ corresponde a função de indicação. A função margem calcula o quanto a média de votos para a relação X, Y da classe correta é maior que a média dos votos para as outras classes. Assim, quanto maior for o valor de margem, maior é a certeza da classificação correta.

Para o cálculo do erro de classificação, podemos definir a função de erro como:

$$PE^* = P_{X,Y}(margem(X, Y) < 0), \quad (12)$$

onde é calculado a probabilidade de X, Y em relação ao valor calculado pela função de margem.

3. Materiais e Métodos

Em geral, a metodologia para o desenvolvimento de um modelo de classificação pode ser dividido nas seguintes etapas: a coleta dos sinais do cérebro através de um EEG; a extração e seleção dos melhores atributos; e por fim a implementação de um classificador a fim de reconhecer padrões nesses dados coletados.

A Figura 1 apresenta um esquemático com todas as etapas que foram desenvolvidas em sua sequência de desenvolvimento.

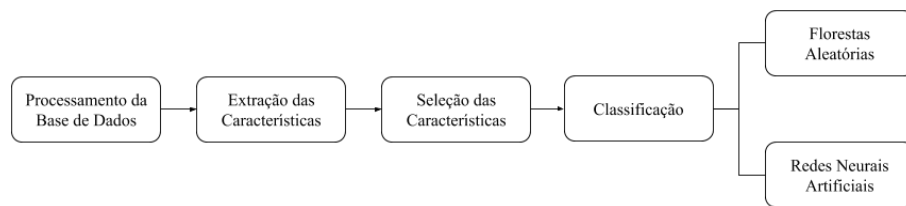


Figura 1. Esquemático com as etapas implementadas durante o desenvolvimento.

3.1. Base de Dados

Para o desenvolvimento deste artigo, foi utilizado a base de dados DEAP [Koelstra et al. 2012]. Os dados foram obtidos por meio de experimentos no qual cada participante era submetido a estímulos audiovisuais na forma de clipes musicais. O estudo contou com 32 participantes, que foram submetidos a 40 experimentos cada. Durante o experimento, uma série de sensores realizavam o monitoramento de diversas medidas, como sinais fisiológicos e atividade cerebral. Especificamente, foram coletadas medidas por meio de um eletroencefalograma (EEG), eletromiografia (EMG) e eletrooculograma (EOG), entretanto somente os dados gerados pelo EEG serão considerados neste trabalho. No EEG foram utilizados 32 canais de acordo com o International 10-20 System. Por fim, o dataset apresenta os valores de *Valence*, *Arousal* e *Dominance*, obtidos por meio de questionários do tipo SAM (Self-assessment Manikin). Resultados empíricos mostram um alto grau de desbalanceamento da base de dados, desta forma, utilizou-se o algoritmo SMOTE para o balanceamento da mesma.

3.2. Pré-processamento dos Dados

Os dados da base de dados foram pré-processados com o objetivo de facilitar a sua manipulação. As medições de EEG realizadas para cada um dos experimentos de cada

participante foram agrupadas em uma única matriz. Para a construção das classes foi utilizado a representação de emoções no espaço tridimensional do modelo emocional PAD. Neste espaço, as dimensões são os valores de *Valence*, *Arousal* e *Dominance* [Liu and Sourina 2012]. Assim, foram mapeadas 8 emoções discretas representadas pela sua posição neste espaço tridimensional [Mehrabian 1995b]. Essas emoções foram: Classe 1 = Proteção; Classe 2 = Satisfação; Classe 3 = Surpresa; Classe 4 = Alegria; Classe 5 = Tristeza; Classe 6 = Despreocupação; Classe 7 = Medo; Classe 8 = Raiva.

Após da definição das classes, a base de dados ainda não está balanceada quanto a distribuição de amostras para cada classe. Assim, a priori a base de dados é separada de forma *one-vs-all*, ou seja, apenas uma classe é considerada por vez e somente então é aplicado o algoritmo SMOTE para realizar o balanceamento dos dados.

3.3. Extração e Seleção de Características

Devido alta quantidade de canais contidos no EEG a utilização de uma técnica de seleção de características se faz necessária. Em [Jenke et al. 2014] é proposto a utilização de coeficientes de Hjorth, criados por [Hjorth 1970], para a classificação de EEG. Estes coeficientes são obtidos pela análise de características descritivas dos sinais de EEG. Os coeficientes de Hjorth podem ser obtidos pelas seguintes equações:

$$Atividade : A_{\xi} = \frac{\sum_{t=1}^T (\xi(t) - \mu)^2}{T} \quad (13)$$

$$Mobilidade : M_{\xi} = \sqrt{\frac{var(\xi \cdot (t))}{var(\xi(t))}} \quad (14)$$

$$Complexidade : C_{\xi} = \frac{M(\xi \cdot (t))}{M(\xi(t))} \quad (15)$$

Após a extração dos coeficientes de Hjorth, foi aplicado o algoritmo t-SNE para se reduzir o número de dimensões e, conseqüentemente, selecionar as melhores características para a representação dos dados no novo hiperplano construído pelo algoritmo. Através de testes empíricos, foi definido que o algoritmo utilizaria somente três dimensões no novo espaço.

3.4. Implementação dos Classificadores

Com o intuito de melhorar o desempenho dos modelos de classificação, o problema foi dividido em 8 subproblemas de classificação binária. Ou seja, ao invés de se utilizar implementações multi classes, o conjunto de dados foi transformado em 8 conjunto de dados, cada um visando a classificação de uma única emoção. Em seguida foi utilizado o algoritmo SMOTE para se obter o balanceamento das classes.

O modelo utilizando uma RNA de Multicamada foi implementada em *Python* utilizando a biblioteca Scikit-Learn [Pedregosa et al. 2011] em *Python*, com o algoritmo *Backpropagation* como algoritmo de aprendizado, em que foram realizados testes empíricos a fim de determinar a melhor arquitetura e parâmetros para a rede. Por fim, foram definidos os seguintes parâmetros:

- Taxa de aprendizado: 0.001;
- Taxa de momentum: 0.9;
- Número máximo de épocas de treinamento: 500;
- Número de camadas escondidas: 2;
- Número de neurônios por camada escondida: 50.

Os testes foram executados 10 vezes por meio de validação cruzada, aonde 80% dos dados foram utilizados para treinamento e 20% como dados de teste. Após realizados os testes, foram contabilizadas as métricas de desempenho do modelo.

Após a implementação do modelo de classificação utilizando RNA, foi implementado o modelo utilizando o algoritmo de Florestas Aleatórias. Para a implementação do algoritmo de Florestas Aleatórias foi definido um número máximo de 1000 árvores que seriam criadas na floresta.

4. Resultados e Discussão

Após as etapas de pré-processamento e implementação dos classificadores utilizando Florestas Aleatórias e RNA foi possível comparar a eficácia a utilização dessas técnicas para classificar emoções em dados de EEG.

Na Tabela 1 são apresentados os percentuais de acurácia obtidos pelo modelo de classificação utilizando o algoritmo de Florestas Aleatórias. O quanto mais próximo de 1, melhor é a eficácia do modelo. Ainda na Tabela 1 é possível observar uma diferença no valor de acurácia entre as classes, onde para a Classe 3 (Surpresa) foi obtido um valor médio de acurácia de 92.5%; por fim, para a Classe 4 (Alegria) o modelo não obteve a mesma capacidade de classificação e obteve um resultado médio de 74.2%.

Tabela 1. Resultados de acurácia para o modelo com Florestas Aleatórias

#	Proteção	Satisfação	Surpresa	Alegria	Tristeza	Despreocupação	Medo	Raiva
Teste 1	0.92	0.84	0.93	0.72	0.85	0.91	0.89	0.89
Teste 2	0.91	0.843	0.92	0.75	0.84	0.91	0.88	0.90
Teste 3	0.94	0.89	0.93	0.74	0.84	0.90	0.88	0.88
Teste 4	0.93	0.85	0.93	0.75	0.84	0.93	0.87	0.90
Teste 5	0.93	0.85	0.92	0.75	0.84	0.93	0.87	0.88
Teste 6	0.93	0.81	0.93	0.73	0.83	0.93	0.87	0.89
Teste 7	0.92	0.84	0.92	0.77	0.85	0.93	0.84	0.90
Teste 8	0.91	0.84	0.93	0.74	0.84	0.95	0.86	0.90
Teste 9	0.92	0.84	0.92	0.74	0.83	0.93	0.87	0.89
Teste 10	0.92	0.83	0.92	0.73	0.86	0.91	0.84	0.88
Média	0.923	0.8433	0.925	0.742	0.842	0.923	0.867	0.891
$\sigma (\pm)$	0.009	0.02	0.005	0.013	0.009	0.014	0.016	0.008

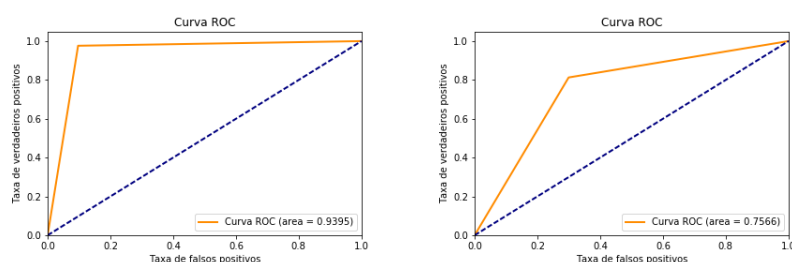
Na Tabela 2 são apresentados os resultados de acurácia obtidos pela RNA de Múltiplas Camadas, onde percebe-se que este modelo apresentou uma maior dificuldade na classificação das emoções. O modelo apresentou resultados significativamente mais baixos quando comparado aos resultados obtidos pelo modelo utilizando Florestas Aleatórias.

As Figuras 2(a) e 2(b) apresentam as curvas ROC (Receiver Operating Characteristic) obtidas pelos modelos utilizando Florestas Aleatórias e RNA de Múltiplas Camadas para a Classe 3 (Surpresa).

As tabelas 3 e 4 apresentam matrizes de confusão com os resultados obtidos pelos modelos com os dados de treinamento para a classificação da Classe 3 (Surpresa).

Tabela 2. Resultados de acurácia para o modelo com RNA de Múltiplas Camadas

#	Proteção	Satisfação	Supresa	Alegria	Tristeza	Despreocupação	Medo	Raiva
Teste 1	0.73	0.63	0.75	0.54	0.61	0.62	0.71	0.66
Teste 2	0.71	0.64	0.76	0.56	0.68	0.70	0.69	0.70
Teste 3	0.72	0.66	0.76	0.60	0.68	0.68	0.72	0.69
Teste 4	0.76	0.67	0.73	0.56	0.69	0.69	0.71	0.64
Teste 5	0.71	0.63	0.72	0.59	0.60	0.69	0.72	0.67
Teste 6	0.74	0.61	0.75	0.54	0.68	0.71	0.65	0.70
Teste 7	0.76	0.62	0.74	0.57	0.65	0.64	0.67	0.71
Teste 8	0.74	0.65	0.76	0.54	0.6	0.71	0.66	0.70
Teste 9	0.73	0.61	0.77	0.55	0.65	0.66	0.70	0.62
Teste 10	0.70	0.63	0.71	0.49	0.68	0.64	0.66	0.62
Média	0.73	0.635	0.745	0.554	0.652	0.674	0.689	0.671
σ (\pm)	0.02	0.02	0.019	0.03	0.03	0.03	0.02	0.03



(a) Curva ROC - Florestas Aleatórias (b) Curva ROC - RNA de Múltiplas Camadas

Figura 2. Resultados obtidos para os dados de treinamento**Tabela 3. Matriz de Confusão para os dados de treinamento (Florestas Aleatórias - Supresa)**

	Positivo	Negativo	Total
Positivo	180 (36.88%) - VP	57 (11.68%) - FN	237
Negativo	42 (8.6%) - FP	209 (42.82%) - VN	251
Total	222	266	488

Tabela 4. Matriz de Confusão para os dados de treinamento (RNA de Múltiplas Camadas - Surpresa)

	Positivo	Negativo	Total
Positivo	166 (34.01%) - VP	71 (14.54%) - FN	237
Negativo	47 (9.63%) - FP	204 (41.8%) - VN	251
Total	213	275	488

A alta complexidade do problema justifica a baixa taxa de acurácia apresentada pelas RNA e favorece a técnica de Florestas Aleatórias, uma vez que esta se baseia na técnica de *ensemble*¹. Em geral, a técnica de Florestas Aleatórias apresentou resultados melhores, com uma média de acurácia para todas as emoções de aproximadamente 87%. Já o modelo que utiliza RNA de Múltiplas Camadas apresentou uma média de acurácia de 64%.

5. Conclusão

Este trabalho teve por objetivo principal explorar a utilização de técnicas de aprendizado de máquina para a classificação de emoções discretas em sinais de EEG. Através da análise das métricas de desempenho acurácia e análise da curva ROC, foi observada uma alta eficácia na utilização de Florestas Aleatórias quando comparada a utilização de RNA de Múltiplas Camadas. Como propostas de continuidade, a investigação de outras

¹Utiliza a combinação de técnicas de aprendizado de máquina para melhorar o desempenho dos modelos de classificação.

técnicas de aprendizado de máquina como arquiteturas *Deep Learning* pode apresentar melhores resultados em vista da sua alta capacidade de modelar problemas complexos.

6. Agradecimentos

Os autores gostariam de agradecer à Universidade Federal de Lavras pelo apoio e por fornecer a estrutura física, e aos integrantes do LABSINE pelo apoio e contribuições.

Referências

- Altman, N. S. (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, 46(3):175–185.
- Barrett, L. F. (2006). Are Emotions Natural Kinds? *Perspectives on Psychological Science*, 1(1):28–58.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.
- Drever, J. (1932). The nature of emotion. *The Journal of Abnormal and Social Psychology*, 27(3):297–301.
- Haykin, S. (2008). *Neural Networks and Learning Machines*, volume 3 of *Neural networks and learning machines*. Prentice Hall.
- He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.
- Hinton, G. E. and Roweis, S. T. (2002). Stochastic neighbor embedding. *NIPS*, pages 833–840.
- Hjorth, B. (1970). EEG analysis based on time domain properties. *Electroencephalography and Clinical Neurophysiology*, 29(3):306–310.
- Izard, C. E. (2007). Basic Emotions, Natural Kinds, Emotion Schemas, and a New Paradigm. *Perspectives on Psychological Science*, 2(3):260–280.
- Jenke, R., Peer, A., and Buss, M. (2014). Feature extraction and selection for emotion recognition from EEG. *IEEE Transactions on Affective Computing*, 5(3):327–339.
- Koelstra, S., M??hl, C., Soleymani, M., Lee, J. S., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A., and Patras, I. (2012). DEAP: A database for emotion analysis; Using physiological signals. *IEEE Transactions on Affective Computing*, 3(1):18–31.
- Kotti, M. and Stylianou, Y. (2017). Effective emotion recognition in movie audio tracks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5120–5124. IEEE.
- Lerner, J. S. and Keltner, D. (2000). Beyond valence: Toward a model of emotion-specific influences on judgement and choice. *Cognition & Emotion*, 14(4):473–493.
- Liu, Y. and Sourina, O. (2012). EEG-based dominance level recognition for emotion-enabled interaction. *Proceedings - IEEE International Conference on Multimedia and Expo*, pages 1039–1044.

- Mauss, I. B. and Robinson, M. D. (2009). Measures of emotion: A review. *Cognition & Emotion*, 23(2):209–237.
- Mehrabian, A. (1995a). Framework for a comprehensive description and measurement of emotional states. *Genetic, social, and general psychology monographs*, 121(1978):339–361.
- Mehrabian, A. (1995b). Framework for a comprehensive description and measurement of emotional states. *Genetic, social, and general psychology monographs*, 121(1978):339–361.
- Mehrabian, A. (1996). Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in Temperament. *Current Psychology*, 14(4):261–292.
- Nugrahaeni, R. A. and Mutijarsa, K. (2017). Comparative analysis of machine learning KNN, SVM, and random forests algorithm for facial expression classification. *Proceedings - 2016 International Seminar on Application of Technology for Information and Communication, ISEMANTIC 2016*, pages 163–168.
- Pedregosa, F., Varoquaux, G., and Gramfort, A., e. a. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Pereira, E. T. and Gomes, H. M. (2016). The role of data balancing for emotion classification using EEG signals. *Digital Signal Processing (DSP), 2016 IEEE International Conference on*, pages 555–559.
- Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178.
- Sammler, D., Grigutsch, M., Fritz, T., and Koelsch, S. (2007). Music and emotion: Electrophysiological correlates of the processing of pleasant and unpleasant music. *Psychophysiology*, 44(2):293–304.
- Van Der Maaten, L. (2014). Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research*, 15:1–21.
- Van Der Maaten, L. and Hinton, G. (2012). Visualizing non-metric similarities in multiple maps. *Machine Learning*, 87(1):33–55.
- Van Der Maaten, L. J. P. and Hinton, G. E. (2008). Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.
- Widrow, B. and Hoff, M. (1960). *Adaptive switching circuits*. Number 4.
- Zhang, F., Meng, H., and Li, M. (2016). Emotion extraction and recognition from music. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2016*, pages 1728–1733. IEEE.
- Zhang, J., Chen, M., Hu, S., Cao, Y., and Kozma, R. (2017). PNN for EEG-based emotion recognition. In *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings*, pages 2319–2323. IEEE.

Estimation of Origin-Destination Matrix from Traffic Counts in the City of Porto with PSO and Taxi Trip Data

Luciano Urgal Pando¹, Ricardo Lüders¹

¹CPGEI - Pós-Graduação em Engenharia Elétrica e Informática Industrial
Universidade Tecnológica Federal do Paraná (UTFPR)
Av. Sete de Setembro, 3165 – 80230-901 – Curitiba – PR – Brasil

lucianourgal@hotmail.com, luders@utfpr.edu.br

Abstract. *The knowledge of urban mobility pattern is important to provide better public services and planning infrastructure. Traffic counts of vehicles installed on streets can be used to estimate flow of vehicles among several origins and destinations in a city. Several techniques are used to estimate origin-destination (O-D) matrices as solution of an optimization problem. Most of them require a priori knowledge about preferential routes. This paper aims to estimate the O-D matrix of urban traffic in a large road network found in the city of Porto in Portugal, using taxi trip data to evaluate preferential routes and PSO for solving the optimization problem. To deal with hundreds of origins and destinations, a clustering algorithm based on density is also applied to group areas of interest. The results show 13 clusters which generate 169 O-D pairs. Estimates are evaluated through three metrics in different time periods of the day.*

1. Introduction

Urban mobility plays an important role in providing good public services for large cities around the world. Since 80's several techniques have been developed to evaluate traffic of vehicles from sensors installed on roads [1]. This information can be used to estimate the flow of vehicles between a pair of origin and destination (O-D) areas. At the same time, vehicles are equipped with Internet connection and GPS technology which provide a huge mass of data useful to understand urban mobility. Moreover, according to [2], there is a strong correlation between traffic of taxis and the general traffic of vehicles passing through a region. This fact represent an advantage over physical sensors as GPS data sources are growing in number and area monitored.

The O-D matrix estimation considers pairs of O-D areas connected by routes defined as a set of links (edges in a graph) in a road network. A single link can be shared among several routes connecting different O-D pairs. Moreover, some links can be monitored by sensors of vehicle counts. The O-D matrix estimation finds the flow of vehicles in each route that generates the vehicle counts measured in each link. This can be defined as an optimization problem whose objective is to minimize the differences between the vehicle counts measured and estimated in each link. In general, this is a multi modal optimization as the number of variables (route flows) is greater than the number of equations (vehicle counts in each link). Extra information is thus necessary to guide the search. There are several techniques to estimate O-D matrices. Some of them are based on statistical methods [3, 4, 5]. Others use mathematical programming [6, 7]. Most of them

needs some extra information to guide the search, particularly about preferential routes to be used in an O-D pair. According to [1], it is surprising that few approaches apply to large road networks.

Our main contribution is to estimate the O-D matrix for a large road network found in Oporto, Portugal. Through that process is possible to notice that the edge probability usage changes throughout the day, and such change influences the estimation results. Also, is possible to notice that a single error measure is not sufficient to fully understand the nature of the errors of an O-D matrix estimate. Several techniques have been used: clustering of origins and destinations, mapping of taxi trips into routes in a graph as well as PSO for solving the optimization problem. In [8], a similar clustering strategy is used. However, travel times are evaluated.

This paper is organized as follows. Section 2 presents the methodology and details of the techniques applied. In Section 3, Section 3.1 shows the outlier detection process, whose importance is to create a more consistent dataset since some days of the year have a abnormal behavior. Section 3.2 shows the spatial clustering process, which is a necessary step for O-D pairs identification, since in a worst case scenario, all map nodes can be viewed as origin and destination points. Finally, section 3.3 shows O-D matrices estimation problem under different scenarios, using sections 3.1 and 3.2 previous work. Finally, conclusions are show in Section 4.

2. Methodology

The data set contains information about the city of Oporto in Portugal whose urban area spans beyond the county boundaries. It is open data available in [9].

The main characteristics are:

- 442 taxi trips collected from 07/01/2013 to 06/30/2014;
- taxi positions are collected every 15 seconds;
- 1,710,671 taxi's trips records;
- 1.8 GB of .csv file.

Nine data attributes are presented in Table 1. POLYLINE, TIMESTAMP and DATATYPE. They identify routes in time and space occurring in weekdays or holidays. The other data (TRIP ID, CALL TYPE, ORIGIN CALL e TAXI ID) are administrative or private information not used in this work.

The data set contains taxi positioning for each trip every 15 seconds in the POLYLINE string. Based on this partial information, a route can be built. However, several paths can be generated and a minimal path algorithm has also been used as 15 seconds causes some information lose. This route is inferred from city map in objects of type nodes and ways through an open map database [10]. At the end, the route information is composed by nodes and edges in a directed graph. Once routes have been identified it is possible to associate traffic counts to each link as virtual traffic sensors [11].

2.1. Outliers

Outliers have been detected by computing the Local Outlier Factor (LOF) for each day in the data set. Proposed by [12], it is measured as the mean ratio of local reachability of a given data register (day) to its k close neighbors. High LOF values indicate outliers.

Table 1. Data set attributes.

Attribute	Type	Description
TRIP ID	String	Single trip ID
CALL TYPE	Char	A (trip launched by a central station) B (ordered directly in taxi stops) C (other)
ORIGIN CALL	Integer	Taxi stop ID, if CALL TYPE = B
TAXI ID	Integer	Driver's ID
TIMESTAMP	Integer	Trip start (seconds).
DATATYPE	Char	B (weekend or other special day) C (a day before type B day) A (other day)
MISSING DATA	Boolean	Acknowledge of full data for POLYLINE
POLYLINE	String	GPS coordinates every 15 seconds, in WGS84 format.

LOF is computed from attributes of each data register. In this work, 20 attributes have been selected for each day. Trip distances were partitioned into 9 quantiles (5, 10, 20, 35, 50, 65, 80, 90, and 95%) as well as 9 quantiles for trip times, plus the number of trips in a day.

2.2. Clustering

Clustering algorithms group n points defined in a d -dimensional space into k clusters. In this article a iterative k-means method is used to find a optimal number of clusters (k). Density based spatial clustering is one of the most common algorithms, such as k-means [13]. Basically, it generates clusters as dense areas of arbitrary shapes. In this article, a density based clustering algorithm is used considering the previously found k value in the iterative k-means method.

2.3. O-D matrix estimation

An O-D matrix contains traffic flow information from source to sink nodes often located into the borders of an area of interest [14]. In this generalized definition, every node can be origin or destination, and huge O-D matrices can be created for large networks. A more basic definition can be found in [15] which represents traffic by a model of gravity. According to [16], origin-destination matrices are usually obtained from expensive field surveys collected in residences or in traffic. However, these surveys are quickly out of date. Therefore, there is a great interest on developing fast and cheap techniques to obtain O-D matrices.

Techniques to estimate O-D matrices based on traffic data counters have begun in 80's using mathematical programming or statistical methods [17]. There are two classes of problems for estimating O-D matrices [18]: Static O-D Estimation (SODE) and Static Traffic Assignment (STA). They can be viewed each other as inverse problems. The goal in SODE is to estimate an O-D matrix from data counters installed in each link. In STA, traffic in each link should be determined from O-D matrix. Both ones need previous knowledge about route choice probabilities.

The O-D matrix estimation is complex due to multiple causes phenomenon [18]. Often there is less equations (traffic counts in each link) than variables (estimated traffic in each O-D pair) and thus multiple solutions arise. An initial O-D trip matrix is then used to find a solution that approaches a current trip pattern which can be determined by previous surveys.

The O-D matrix estimation can be formulated as an optimization problem according to (1) - see [1, 6].

$$\begin{aligned} \min_{\mathbf{T}} \sum_{i=1}^m (v_i - \bar{v}_i)^2 \\ \text{s.t. } \sum_{j=1}^n p_{ij} T_j = v_i, \quad i = 1, \dots, m \end{aligned} \quad (1)$$

The term $(v_i - \bar{v}_i)$ in (1) is the difference between the estimated and counted traffic in each link i , respectively, T_j is the estimated traffic per hour in the O-D pair j expressed as a positive real number, and p_{ij} is the proportion of traffic T_j passing through link i (O-D pairs are numbered from 1 to n).

In this work, the solution of the optimization problem (1) is obtained by PSO (Particle Swarm Optimization) [19]. In this case, each individual is coded as a real value vector of n elements T_j in order to find a solution which minimizes (1).

3. Results

The geographical region of the city of Porto considered in this work is bounded by latitude 41.0374522 to 41.3142094 and longitude -8.7314937 to -8.3749317 .

3.1. Outlier detection

The software package WEKA [20] was used to compute LOF values for each of 364 days. The histogram of LOF is shown in Fig. 1 with mean 1.024, standard deviation 0.048, and minimum and maximum values 0.997 and 1.378, respectively. LOF values greater than 1.168 (mean plus three standard deviations) were considered outliers according to Fig. 1. It turns out that 10 from 364 days were considered outliers. All data from these outlier days have been dismissed as they correspond to abnormal traffic behaviors.

3.2. Clustering

In large road networks we are interested in traffic between city areas rather than point locations. Origins and destinations of taxi calls recorded in the data set are then clustered by geographical position, reducing the number of O-D pairs. This is accomplished by a clustering algorithm [8].

In order to make the spatial clustering, a new dataset needs to be done from the original data. This new dataset contains only locations (latitude and longitude). For each instance of the original dataset, one instance is generated for origin, and another for destination. Clustering with all points in the dataset could not be performed because of memory limitations, so the dataset has been reduced to 10% of its original size. Another adaptation was made as a post processing, with the exclusion of small clusters (less than

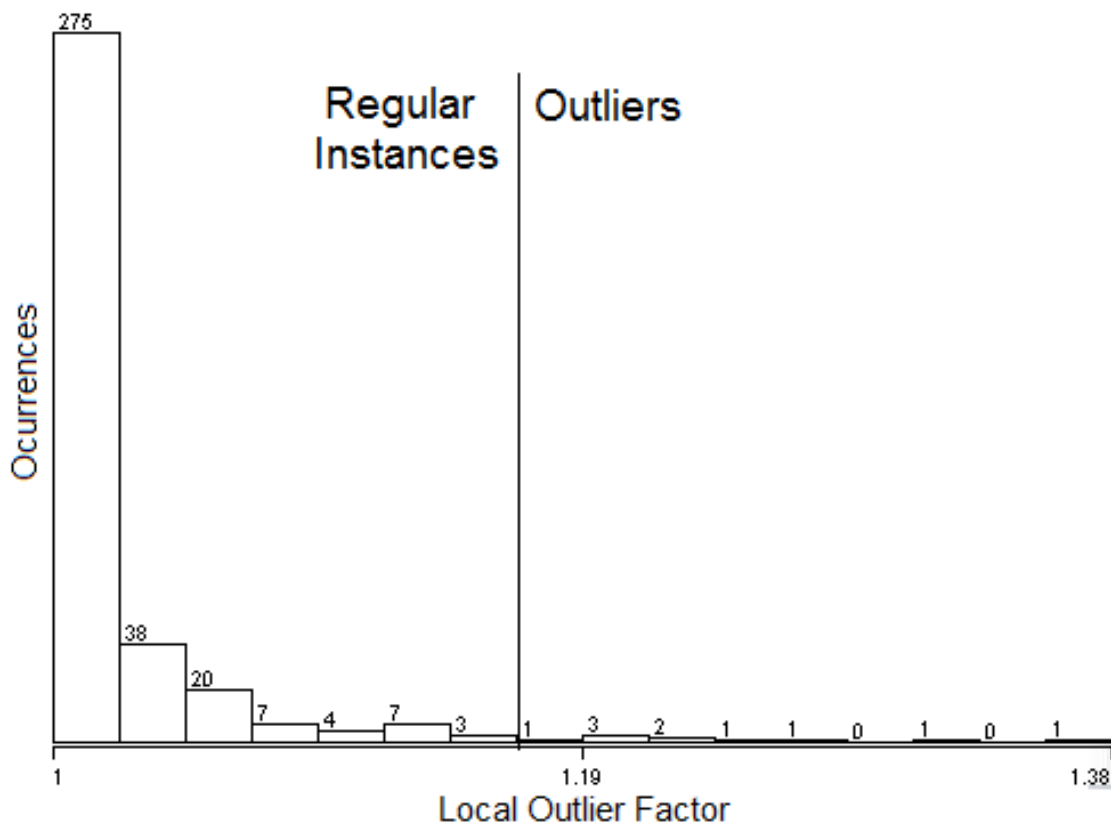


Figure 1. Histogram of LOF.

0.25km²), in order to decontaminate the OD matrix without considerable loss of information, by joining such clusters to the neighbor cluster with the closest density.

The iterative method *CascadeSimpleKMeans* of WEKA was used. This method identify an ideal number of clusters with considerable processing time. The result was $k = 14$ clusters. Then the density based clustering method *MakeDensityBasedClusterer* from WEKA was set with $k = 17$ clusters in order to have a final number of clusters close to 14, after post processing. The number of clusters was then reduced to $k = 13$ after dismissing the three smallest clusters plus the cluster of remaining points that do not fit in any other cluster. The cluster centroids numbered from c_0 to c_{12} can be seen in in Figure 2.

Once clusters are obtained, traffic flows from different taxi trips can be aggregated for the same O-D pair in a total of 169 O-D pairs which cannot be shown due to lack of space. The probability of link usage was calculated for each O-D pair by mapping routes as a set of links in the map of the city [10]. This information is required for the O-D matrix estimation algorithms.

Table 2 summarizes the information obtained for each cluster. It can be seen the total cluster area, the number of taxi trips originated in the cluster and finished as well. Note that small areas can be origin or destination of lots of trips as well as big areas can concentrate few trips.

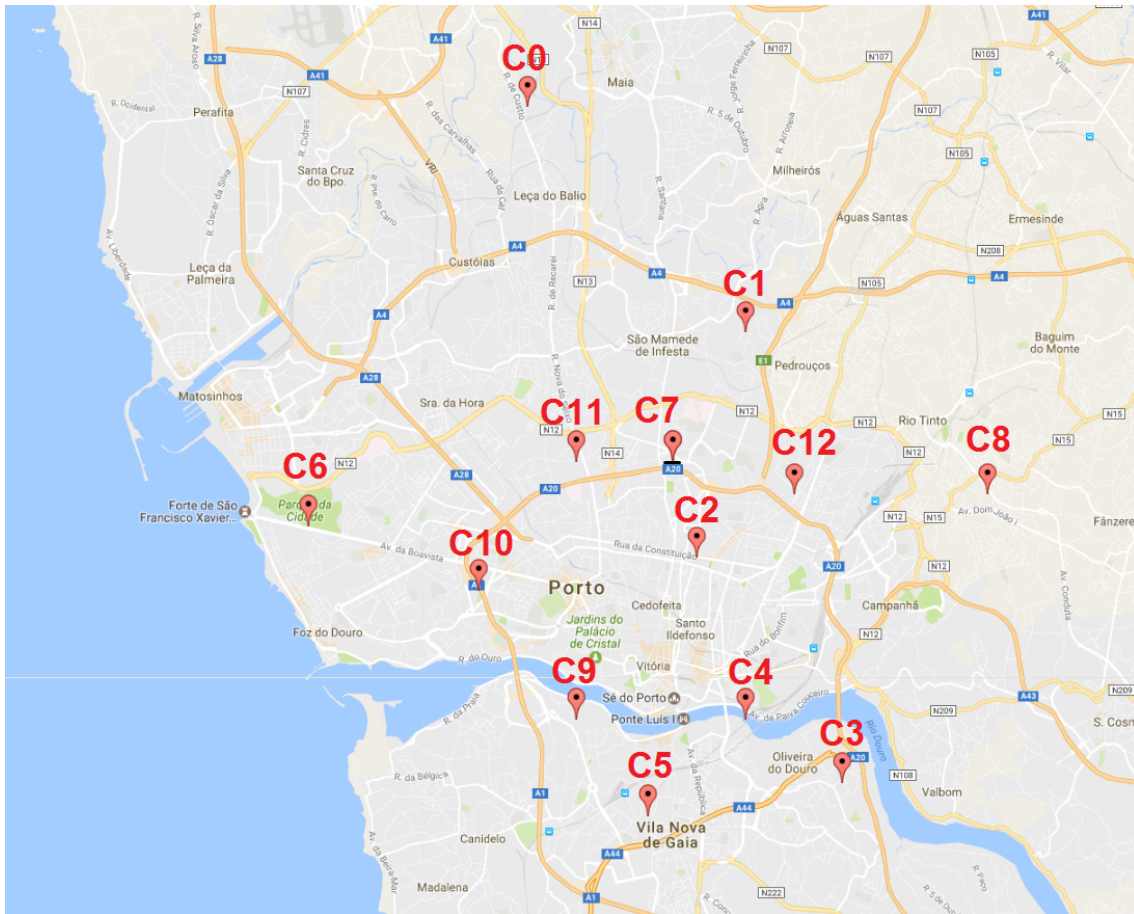


Figure 2. Clustering by density (13 clusters).

3.3. O-D matrix estimation

As result of route generation for each taxi trip, 82% of GPS positioning data could be connected, i.e., 18% of taxi GPS positioning data remained isolated. Based on traffic of each link, the proportion p_{ij} of each route connecting two O-D pairs (i, j) and passing through the link has been obtained for 8:00 to 9:00 and 16:00 to 17:00 to capture traffic patterns both in the morning and in the afternoon.

The parameters of PSO have been selected from sets $w_{initial} \in \{0.3, 0.4\}$, $w_{final} \in \{0.2, 0.15, 0.1\}$, $c1 \in \{0.2, 0.3, 0.4\}$, $c2 \in \{0.2, 0.3, 0.4\}$ and $(s/t) \in \{(10/6), (10/4), (10/2)\}$. An iterative procedure has tried all combinations to find the best one. This procedure has considered 2,000 virtual sensors and maximum 18,000 fitting evaluations for each PSO execution. A total of five PSO executions have been made. The solution was coded into a vector of 169 real numbers representing vehicle counts per hour for each O-D pair. The final PSO parameters are shown in Table 3.

Tests have been accomplished to compute O-D matrix estimations for different periods of time during the day. They are named pXtY with $X \in \{8, 16\}$ and $Y \in \{8, 10, 14, 16\}$ meaning that p_{ij} was obtained for time X but the observed traffic for time Y. A total of 7,500 virtual sensors have been used in 15 PSO executions for each test. Each execution has taken approximately 60 minutes to complete in a AMD R7 1700 3.0 GHz with 16 GB 3.0 Ghz RAM.

Table 2. Cluster information.

Cluster	Area (km^2)	# of Origins	# of Destinations
c0	47	2258	12146
c1	12	42494	25530
c2	3	60532	108054
c3	43	30738	41511
c4	6	115528	83467
c5	15	375361	245454
c6	36	101930	77865
c7	3	28781	42170
c8	120	4649	12987
c9	9	154907	217807
c10	17	154685	209749
c11	7	60114	69462
c12	7	87707	72484

Table 3. PSO parameters.

$w_{initial}$	w_{final}	c1	c2	s	t
0.4	0.2	0.4	0.4	3750	1500

The results of O-D matrix estimation were evaluated by means of the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) computed between the solution provided by PSO and the real one collected from taxi trip data in different periods of the day. For link adequacy between observed and estimated counts, the GEH metric was used [21].

In Table 4, the best results are marked in bold according to the Mann-Whitney test. Better results are obtained when the same time is used for both observed traffic and route proportion, i.e, for p8t8 and p16t16. As far as the time difference between observed traffic and route proportion increases, worst results are obtained (except for p8t10 and p8t14 using MAE). The same result can be seen in Figs. 3, 4 and 5 with box plots for GEH, MAE, and RMSE, respectively.

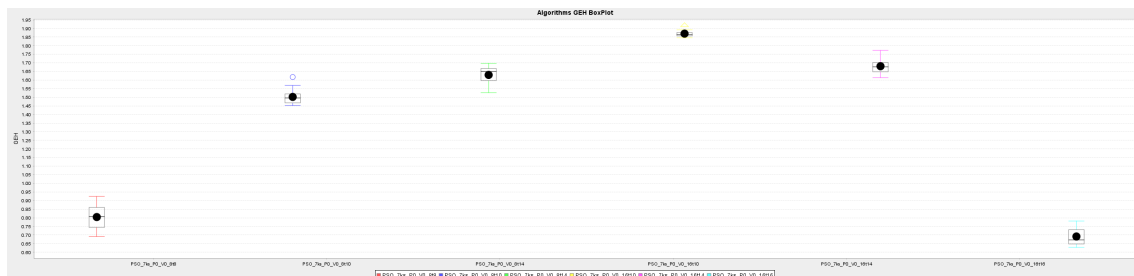
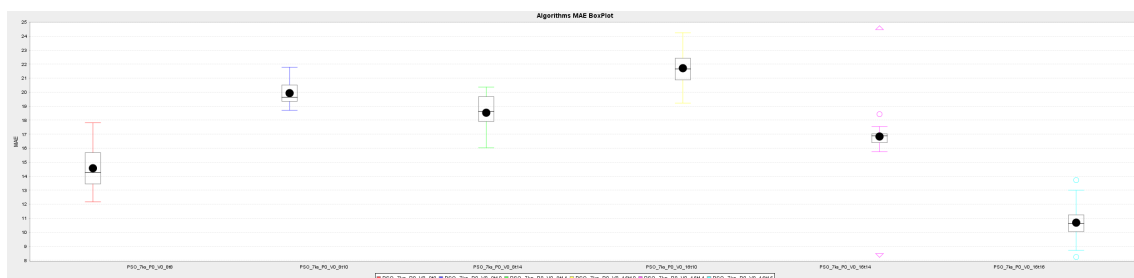


Figure 3. GEH box plot for p8t8, p8t10, p8t14, p16t10, p16t14, p16t16 (from left to right).

Figure 6 shows the correspondence between estimated and observed link values for p8t8 and p8t14. In an ideal situation, the graphics would be a straight line with all links with identical observed and estimated counts. Comparing the results for p8t8 and

Table 4. O-D matrix estimation MAE, RMSE and GEH.

		min	median	max
MAE	p8t8	12.16822	14.26007	17.81617
	p8t10	18.70374	19.64257	21.7841
	p8t14	16.02554	18.63391	20.36534
RMSE	p8t8	65.27992	92.84879	131.90733
	p8t10	104.81245	119.99981	141.54077
	p8t14	112.13252	137.88567	151.84182
GEH	p8t8	0.69083	0.80668	0.92426
	p8t10	1.45203	1.49597	1.60285
	p8t14	1.52702	1.65019	1.69778
MAE	p16t16	8.05361	10.62367	13.53915
	p16t14	15.11409	16.87871	18.85367
	p16t10	19.22185	21.65967	24.2372
RMSE	p16t16	59.90685	88.54069	136.18761
	p16t14	109.18068	123.95497	159.93604
	p16t10	122.23695	134.80304	163.86844
GEH	p16t16	0.62821	0.67132	0.78136
	p16t14	1.61427	1.67686	1.77245
	p16t10	1.84536	1.86378	1.92567

**Figure 4. MAE box plot for p8t8, p8t10, p8t14, p16t10, p16t14, p16t16 (left to right).**

p8t14, it is noticeable that a scatter effect increases for p8t14 indicating a higher GEH value. Finally, it is possible to observe that most of the links have counts below 170 with more concentration below 50.

4. Conclusion

This work has presented a strategy for estimating the O-D matrix for traffic in Oporto, Portugal. Although there are many techniques to estimate such matrix, the presented one is suitable to large road networks. A clustering algorithm is used to group origins and destinations of taxi trips into broad areas of interest. This way, 13 clusters have been identified and 169 O-D pairs have been defined. The O-D matrix estimates are obtained by solving an optimization problem with PSO using previous route preferences set according to traffic generated by taxi trips in the morning and in the afternoon. Tests are then accomplished for estimating O-D matrices in time periods of the day other than the ones used to generate route preferences. The results show that better O-D estimates are obtained for traffic and route preferences obtained in the same period of the day (morning or af-

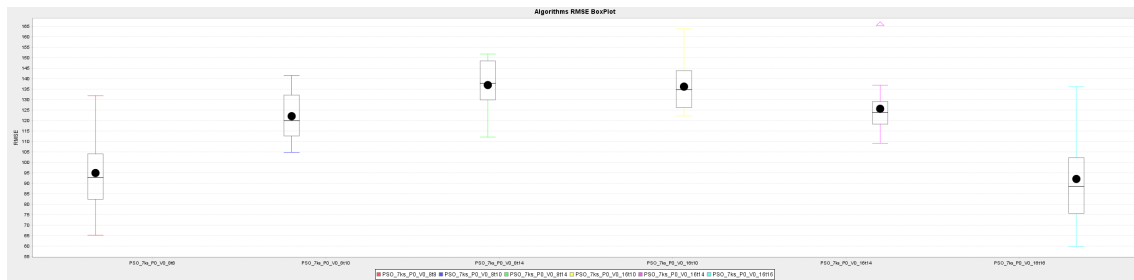


Figure 5. RMSE box plot for p8t8, p8t10, p8t14, p16t10, p16t14, p16t16 (left to right).

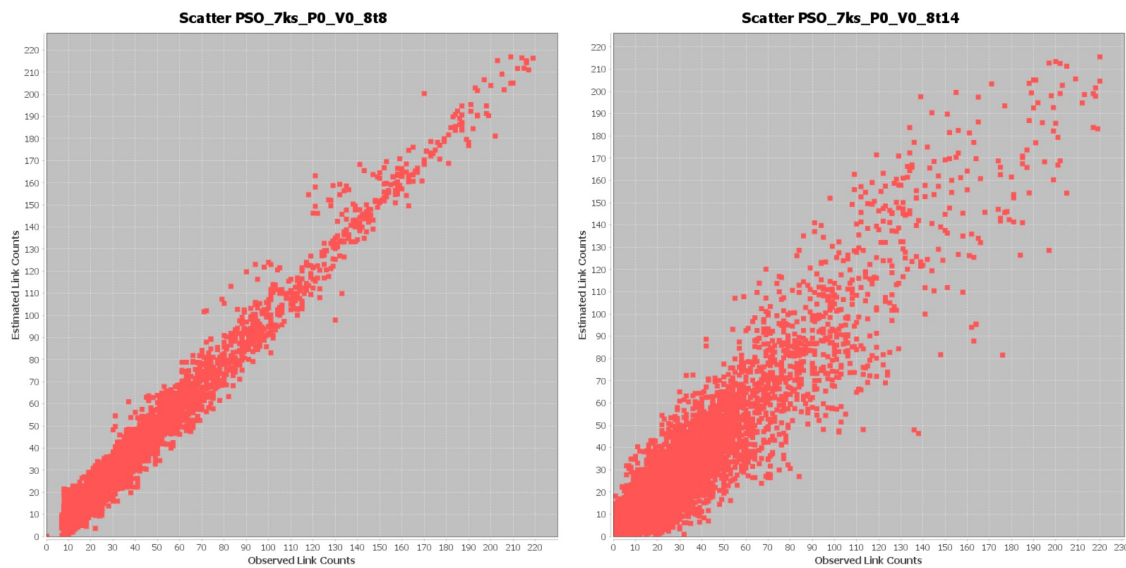


Figure 6. Estimated link counts (vertical axis) versus observed link counts (horizontal axis) for p8t8 and p8t14 (left to right).

ternoon), although they are affected by the performance evaluation metric (RMSE, MAE or GEH). Another conclusion is that each of the metrics is sensible to one kind of error in such a way that they are complementary, and O-D matrix estimation results evaluation is not a simple task. For future work, other estimation techniques should be compared to PSO performance, such as General Least Squares, Bayesian inference, maximum likelihood and mathematical programming. The number and location of traffic sensors should also be considered as the quality of O-D matrix estimation is quite affected by the amount of information provided.

References

- [1] S. Bera and K. Rao, “Estimation of origin-destination matrix from traffic counts: the state of the art,” *European Transport*, v. 49, pp. 3–23, 2011.
- [2] C. F. Daganzo and L. J. Lehe, “Traffic flow on signalized streets,” *Transportation Research Part B: Methodological*, vol. 90, pp. 56–69, 2016. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0191261515302423>

- [3] E. Cascetta, “ESTIMATION OF TRIP MATRICES FROM TRAFFIC COUNTS AND SURVEY DATA : A GENERALIZED LEAST SQUARES ESTIMATOR,” *Transportation Research Part B*, vol. I, no. 415, pp. 289–299, 1984.
- [4] M. J. Maher, “INFERENCES ON TRIP MATRICES FROM OBSERVATIONS ON LINK VOLUMES : A BAYESIAN STATISTICAL APPROACH,” *Transportation Research Part B*, vol. 17, no. 6, 1983.
- [5] H. Spiess, “A MAXIMUM LIKELIHOOD MODEL FOR ESTIMATING ORIGIN-DESTINATION MATRICES,” *Transportation Research Part B*, vol. 21, no. 5, pp. 395–412, 1987.
- [6] H. D. Sherali, R. Sivanandan, and A. G. Hobeika, “A linear programming approach for synthesizing origin-destination trip tables from link traffic volumes,” *Transportation Research Part B: Methodological*, vol. 28, no. 3, pp. 213 – 233, 1994. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0191261594900086>
- [7] H. Yang and J. Zhou, “Optimal traffic counting locations for origin-destination matrix estimation,” *Transportation Research Part B: Methodological*, vol. 32, no. 2, pp. 109–126, 1998.
- [8] L. M. Matias, J. Gama, M. Ferreira, J. M. Moreira, and L. Damas, “Time-evolving od matrix estimation using high-speed gps data streams,” *Expert Systems with Applications*, vol. 44, pp. 275–288, 2016.
- [9] UCI, “Dataset de viagens de txi de porto, portugal,” <https://archive.ics.uci.edu/ml/datasets/Taxi+Service+Trajectory+-+Prediction+Challenge,+ECML+PKDD+2015#>, 2015.
- [10] OpenStreetMap, “Base de dados geográficos,” <https://www.openstreetmap.org/>, 2017.
- [11] J. C. Herrera, D. B. Work, R. Herring, X. J. Ban, Q. Jacobson, and A. M. Bayen, “Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment,” *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 568–583, 2010. [Online]. Available: <http://dx.doi.org/10.1016/j.trc.2009.10.006>
- [12] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying Density-Based Local Outliers,” *Proceedings of the 2000 Acm Sigmod International Conference on Management of Data*, pp. 1–12, 2000. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.8948>
- [13] S. P. Lloyd, “Least Squares Quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [14] K. Teknomo and P. Fernandez, “A theoretical foundation for the relationship between generalized origin-destination matrix and flow matrix based on ordinal graph trajectories,” *Journal of Advanced Transportation*, vol. 48, no. 6, pp. 608–626, 2014.
- [15] A. M. Voorhees, “A general theory of traffic movement,” *Transportation*, vol. 40, no. 6, pp. 1105–1116, 2013. [Online]. Available: <http://link.springer.com/10.1007/s11116-013-9487-0>

- [16] A. Chen, P. Chootinan, and W. W. Recker, “Examining the Quality of Synthetic Origin Destination Trip,” *Journal of Transportation Engineering*, vol. 131, no. July, pp. 506–513, 2005.
- [17] H. S. ren and L. H. tsung, “A generalized sensor location model for the estimation of network origin destination matrices,” *Transportation Research part C*, vol. 40, pp. 93–110, 2014. [Online]. Available: <http://dx.doi.org/10.1016/j.trc.2014.01.004>
- [18] M. Bierlaire, “The total demand scale : a new measure of quality for static and dynamic origin destination trip tables,” *Transportation Research Part B*, vol. 36, pp. 837–850, 2002.
- [19] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948 vol.4, 1995.
- [20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: An update,” *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: <http://doi.acm.org/10.1145/1656274.1656278>
- [21] T. Ma and B. Abdulhai, “Genetic algorithm-based optimization approach and generic tool for calibrating traffic microscopic simulation parameters,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 1800, pp. 6–15, 2002.

Transferência de aprendizagem via método de regressão por vetores suporte

Rodolfo S. Simões¹, Patrícia R. Oliveira¹, Clodoado A. M. Lima¹

¹Escola de Artes, Ciências e Humanidades - Universidade de São Paulo
Av. Arlindo Bettio, 1000, Ermelino Matarazzo, Sao Paulo-SP

simoerodolfo@usp.br, proliveira@usp.br, c.lima@usp.br

Resumo. *O principal pressuposto em muitos algoritmos de aprendizagem de máquina é que os dados de treinamento e os dados de teste devem estar no mesmo espaço de características e devem seguir a mesma distribuição de probabilidade. Em situações que não são verdadeiras, técnicas de transferência de aprendizagem tornam-se valiosas, uma vez que tal abordagem permite que domínios, tarefas e distribuições de probabilidade consideradas nas fases de treinamento e teste sejam diferentes. Neste trabalho, apresenta-se uma abordagem de transferência de aprendizagem indutiva. Tal proposta baseia-se em uma variação do método de Regressão por Vetores Suporte adaptado para transferência de aprendizagem, a qual é alcançada ao aproximar os modelos gerados separadamente para cada tarefa em questão. Resultados experimentais mostram que o método proposto apresenta bom desempenho quando aplicado a alguns conjuntos de dados de benchmark e a problemas reais.*

Abstract. *A major assumption in many machine learning and data mining algorithms is that the training and test data must be in the same feature space and have the same distribution. In such situations where these assumptions are not true, transfer learning techniques become valuable, since such an approach allows domains, tasks, and probability distributions considered in the training and testing phases to be different. In this paper we present an inductive transfer learning approach. This proposal is based on a variation of the Support Vector Regression method for learning transfer, which is achieved by approaching the separately generated models for each task in question. Experimental results show that the proposed method performs well when applied to some benchmark datasets and to real problems.*

1. Introdução

Técnicas de mineração de dados e aprendizagem de máquina, incluindo métodos de classificação, regressão e agrupamento, têm alcançado sucesso significativo em muitas áreas do conhecimento. Entretanto, muitos desses métodos funcionam bem apenas sob a suposição de que os dados de treinamento e teste são oriundos do mesmo espaço de características e seguem a mesma distribuição de probabilidade [Wang and Li 2014]. Quando ocorrem alterações na distribuição dos dados, muitos modelos estatísticos precisam ser reconstruídos a partir da coleta de novos dados de treinamento. No entanto, em muitas aplicações do mundo real, é custoso ou impossível recoletar os dados necessários para

reconstruir esses modelos. Em tais situações, a transferência de aprendizagem¹ (conhecida também como transferência de conhecimento) entre domínios seria desejável, uma vez que tal abordagem permite que domínios, tarefas e distribuições de probabilidade consideradas nas fases de treinamento e teste sejam diferentes [Pan and Yang 2010].

Como exemplo, é possível considerar que aprender a reconhecer maçãs pode ajudar a reconhecer pêras ou que o conhecimento adquirido para tocar violão pode ajudar no processo de aprendizagem para tocar guitarra. De maneira geral, o desenvolvimento de abordagens de transferência de aprendizagem foi motivado pelo fato de que é possível aplicar o conhecimento adquirido anteriormente para resolver novos problemas de forma mais rápida e com melhores soluções. Pode-se encontrar muitos outros exemplos em áreas de conhecimento em que a transferência de aprendizagem agrega benefícios. Dentre estes, destaca-se o problema de classificação de sentimentos, em que a tarefa é classificar automaticamente comentários sobre um determinado produto, como uma câmera fotográfica, em positivo ou negativo. Para essa tarefa de classificação, inicialmente é preciso coletar muitos comentários sobre o produto e após isso, treinar um classificador para os dados coletados com seus respectivos rótulos. Uma vez que a distribuição de probabilidade dos comentários sobre outros produtos pode ser muito diferente, para que classificador mantenha um desempenho satisfatório, é necessário refazer o processo para cada produto. Para reduzir esse esforço, é possível adaptar um modelo de classificação que foi treinado em alguns produtos para ajudar a aprendizagem de modelos de classificação para outros.

De maneira geral, abordagens de transferência de aprendizagem têm se mostrado promissoras para combinar o conhecimento previamente obtido em tarefas relacionadas em um único modelo preditivo, seja este para classificação, regressão ou agrupamento [Pan and Yang 2010]. Neste trabalho, são investigadas situações em que a técnica melhora o poder preditivo de um modelo de regressão baseado em vetores suporte (SVR)². O presente trabalho é organizado da forma como segue. A Seção 2 apresenta uma descrição da abordagem de transferência de aprendizagem, bem como algumas definições e notações. A Seção 3 apresenta o método SVR, a sua versão adaptada e a proposta deste trabalho para transferência de aprendizagem. Na Seção 4 são discutidos os resultados experimentais obtidos. Considerações finais são apresentadas na Seção 5.

2. Transferência de aprendizagem

Define-se transferência de aprendizagem como sendo a habilidade de um sistema reconhecer e aplicar o conhecimento aprendido em tarefas anteriores para a solução de novos problemas relacionados [Pan and Yang 2010]. Portanto, o objetivo dessa abordagem consiste em extrair o conhecimento obtido por um modelo a partir de uma ou mais tarefas fontes e aplicá-lo em uma tarefa destino. Uma das premissas para a utilização da técnica de transferência de aprendizagem é que os domínios de origem e de destino devem ser relacionados. O estudo desenvolvido em [Tan et al. 2015] aponta que tal relação pode ser concretizada na forma de instâncias [Bickel et al. 2009] ou características [Satpal and Sarawagi 2007]. Se nenhuma relação direta for encontrada, a transferência forçada não vai funcionar, resultando em nenhuma melhora ou piorando o desempenho no domínio de destino [Fitzgerald and Thomaz 2015].

¹Do original, em inglês, *Transfer Learning*.

²Do original, em inglês, *Support Vector Regression*.

2.1. Notações e definições

Um domínio D consiste em dois componentes: um espaço de características χ e uma distribuição de probabilidade marginal $P(\mathbf{X})$, em que $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in \chi$. Por exemplo, se o problema de aprendizagem é a classificação de documentos de texto e cada termo nesses documentos é considerado uma característica binária, então χ é o espaço de todos os vetores de termos, \mathbf{x}_i é o i -ésimo vetor de termos correspondente a algum documento e \mathbf{X} é uma amostra em particular. Em geral, se dois domínios são diferentes, então estes possuem espaços de características ou distribuições de probabilidade diferentes.

Dado um domínio específico, $D = \{\chi, P(\mathbf{X})\}$, uma tarefa (denotada por $T = \{Y, f(\cdot)\}$) consiste em dois componentes: um espaço de rótulos Y e uma função de predição $f(\cdot)$, que pode ser aprendida a partir dos dados de treinamento, que, por sua vez, consistem em pares (\mathbf{x}_i, y_i) , tal que $\mathbf{x}_i \in \mathbf{X}$ e $y_i \in Y$. A função $f(\cdot)$ pode ser usada para prever o rótulo correspondente a uma nova instância \mathbf{x} , $(f(\mathbf{x}))$. No exemplo de classificação de documentos, Y é o conjunto de todos os rótulos possíveis, podendo ser {"Positivo", "Negativo"} para uma tarefa de classificação binária.

Então, dado um domínio de origem D_S , uma tarefa de aprendizado de origem T_S , um domínio de destino D_T e uma tarefa de aprendizado de destino T_T , a transferência de aprendizagem tem o objetivo de ajudar a melhorar o aprendizado da função preditiva do destino $f_T(\cdot)$ no domínio D_T , usando o conhecimento de D_S e T_S , em que $D_S \neq D_T$ ou $T_S \neq T_T$ [Pan and Yang 2010]. Caso $D_S = D_T$ e $T_S = T_T$, o problema torna-se um problema tradicional de aprendizagem de máquina.

Para simplificar a notação utilizada neste trabalho, denota-se o domínio de origem como $D_S = \{(\mathbf{x}_{S1}, y_{S1}), \dots, (\mathbf{x}_{Sn}, y_{Sn})\}$, em que $\mathbf{x}_{Si} \in \chi_S$ é uma instância e $y_{Si} \in Y_S$ é o valor do rótulo correspondente. No exemplo de classificação de documentos, D_S pode ser um conjunto de vetores de termos com seus respectivos rótulos associados. Similarmente, $D_T = \{(\mathbf{x}_{T1}, y_{T1}), \dots, (\mathbf{x}_{Tn}, y_{Tn})\}$, em que $\mathbf{x}_{Ti} \in \chi_T$ é uma instância e $y_{Ti} \in Y_T$ é o valor do rótulo de classe correspondente. Em geral, $0 \leq n_T \ll n_S$, em que n_T e n_S são o número de instâncias nos domínios de destino e de origem, respectivamente.

3. Transferência de aprendizagem em modelos de regressão por vetores suporte

Nesta seção serão apresentados os métodos de aprendizagem de máquina considerados neste trabalho. A Seção 3.1 descreve as Máquinas de Vetores Suporte (SVM)³, que servem como embasamento teórico para o método de Regressão por Vetores Suporte, apresentado na Seção 3.2. A Seção 3.3 introduz o método Regressão por Vetores de Suporte Adaptado para Transferência de Aprendizagem (ATL)⁴, que utiliza, em sua formulação, informações de um conjunto de dados auxiliar.

3.1. Máquinas de Vetores Suporte (SVM)

Pode-se dizer que SVM's são modelos de classificação lineares baseados no paradigma de aprendizagem supervisionado. Por se tratar de um modelo linear, as classes analisadas por SVM's são separadas por um hiperplano. Para tal, busca-se maximizar a

³Do original, em inglês, *Support Vector Machine*.

⁴Do original, em inglês, *Adapted to Transfer Learning*.

distância entre os pontos mais próximos da reta, em cada classe, e a própria reta. A esta distância dá-se o nome de margem [Burges 1998].

Para fins didáticos, o problema de classificação pode ser tomado como um problema de duas classes, sem perda de generalidade [Gunn et al. 1998]. Tal tarefa consiste, portanto, em separar duas classes por uma função que é induzida a partir das instâncias de treinamento com o objetivo de produzir um classificador que generaliza bem para dados desconhecidos. Observando o exemplo da Figura 1, nota-se que há vários classificadores lineares possíveis que podem separar as instâncias disponíveis sem nenhum erro, mas há somente um que maximiza a margem, ou seja, maximiza a distância entre o classificador e a instância mais próxima de cada classe. Esse classificador é chamado de hiperplano de separação ótimo, pois, ao menos intuitivamente, espera-se que este generalize melhor que os demais hiperplanos para dados desconhecidos [Lima 2004].

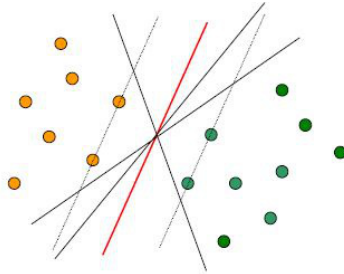


Figura 1. Hiperplano de separação ótimo (em vermelho), tendo como suporte dois outros hiperplanos (linhas tracejadas), os quais passam por alguns pontos para ambas as classes. Estes pontos são chamados de vetores suporte [Lima 2004].

3.2. Regressão por Vetores Suporte (SVR)

O método de vetores suporte pode ser aplicado ao problema de regressão, mantendo-se todas as principais propriedades que caracterizam o algoritmo de margem máxima [Cristianini and Shawe-Taylor 2000]. A extensão de SVM para o tratamento de problemas de regressão se dá pela introdução de uma função de perda [Smola et al. 1996].

Para o problema de regressão, ocorre uma reversão de propósito, quando comparado à formulação de problemas de classificação binária. Ou seja, em vez de forçar as instâncias a se posicionarem o mais distante possível do hiperplano ótimo, no espaço de características, o objetivo agora é conduzi-las a se posicionarem o mais próximo possível do hiperplano, não importando de que lado deste a instância se encontre. Embora os propósitos da otimização sejam opostos, o objetivo de ambas as formulações é o mesmo: maximizar a capacidade de generalização do modelo preditivo [Lima 2004].

3.2.1. Formulação da técnica SVR

Dado um conjunto de instâncias de treinamento $\{(x_i, y_i), i = 1, \dots, l\}$, em que $x_i \in \mathbb{R}^n$ e $y_i \in \mathbb{R}$, o objetivo do modelo de regressão linear é aproximar esse conjunto por meio da função:

$$f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b. \quad (1)$$

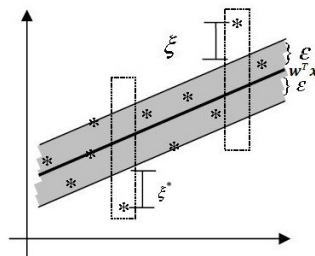
Para isso, o método SVR linear deve encontrar um modelo \mathbf{w} , tal que $\mathbf{w}^T \mathbf{x}_i$ seja próximo do valor alvo y_i . Isso é solucionado pela regularização do problema de otimização a seguir:

$$\min_{\mathbf{w}, \xi^*, \xi} = \frac{1}{2}(\mathbf{w}^T \mathbf{w}) + C(\sum_{i=1}^l \xi_i + \sum_{i=1}^l \xi_i^*). \quad (2)$$

Na Equação (2), o termo $(\mathbf{w}^T \mathbf{w})$ regulariza a complexidade do modelo, $C > 0$ é um parâmetro definido pelo usuário e ξ e ξ_i^* são variáveis de folga representando as restrições dos dois lados do hiperplano (veja Figura 2), conforme as equações:

$$\begin{cases} y_i - (\mathbf{w} \cdot \mathbf{x}_i) - b \leq \epsilon + \xi_i \\ (\mathbf{w} \cdot \mathbf{x}_i) + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, \quad i = 1, \dots, l. \end{cases} \quad (3)$$

Figura 2. Ilustração de vetores suporte para o problema de regressão com as variáveis de folga ξ e ξ^* nos dois lados do hiperplano [Lima 2004].



A função de perda ϵ -insensível para o método SVR, e associada a (\mathbf{x}_i, y_i) , é definida por:

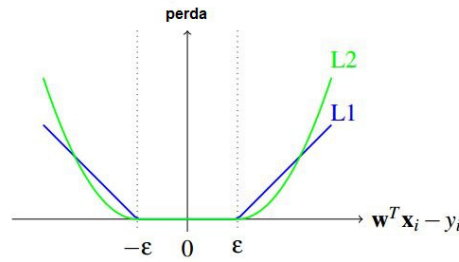
$$L_{\epsilon\text{-insensível}}(f(\mathbf{x}) - y) = \begin{cases} 0 & \text{se } |f(\mathbf{x}) - y| < \epsilon \\ |f(\mathbf{x}) - y| - \epsilon & \text{caso contrário.} \end{cases} \quad (4)$$

Já a função de perda ϵ -quadrática para o método SVR é dada por:

$$L_{\epsilon\text{-quadrática}}(f(\mathbf{x}) - y) = \begin{cases} 0 & \text{se } |f(\mathbf{x}) - y| < \epsilon \\ (|f(\mathbf{x}) - y| - \epsilon)^2 & \text{caso contrário.} \end{cases} \quad (5)$$

Tais funções garantem que a perda é zero se $|\mathbf{w}^T \mathbf{x}_i - y_i| = |f(\mathbf{x}_i) - y_i| \leq \epsilon$. Isso significa que o valor alvo real y_i encontra-se dentro de um intervalo esperado em torno de $\mathbf{w}^T \mathbf{x}$. As duas funções apresentadas nas Equações (4 e 5) são também conhecidas como funções de perda L_1 e L_2 , respectivamente. Uma ilustração dessas funções é apresentada na Figura 3. Neste trabalho, será utilizada a medida de erro quadrático médio (MSE)⁵ como função de erro, que é diretamente modelada pela função de perda L_2 . Assim, as

Figura 3. Funções de perda L_1 e L_2 [Lima 2004].



equações ao longo deste trabalho assumem que a função de perda L_2 , ou ϵ -quadrática, é aplicada.

Utilizando a função de perda ϵ -quadrática, a formulação dual do modelo de regressão assume a seguinte configuração:

$$\max_{\alpha, \alpha^*} \left[-\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(\mathbf{x}_i \cdot \mathbf{x}_j) + \sum_{i=1}^l \alpha_i^*(y_i - \epsilon) - \alpha_i(y_i + \epsilon) \right] - \frac{1}{2C} \sum_{i=1}^l (\alpha_i^2 + (\alpha_i^*)^2). \quad (6)$$

Fazendo a seguinte substituição $\beta_i = \alpha_i - \alpha_i^*$ na Equação (6), viabiliza-se uma simplificação via condições de KKT, em que:

$$\alpha_i \alpha_i^* = 0 \quad i = 1, \dots, l, \quad (7)$$

o que implica em $\beta_i^2 = \alpha_i^2 + (\alpha_i^*)^2$ e $|\beta_i| = |\alpha_i - \alpha_i^*| = \alpha_i + \alpha_i^*$. Com isso, pode-se definir o problema dual considerando a seguinte função de perda L_2 :

$$\min_{\beta} = \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \beta_i \beta_j (\mathbf{x}_i \cdot \mathbf{x}_j) - \sum_{i=1}^l \beta_i y_i + \frac{1}{2C} \sum_{i=1}^l \beta_i^2 + \sum_{i=1}^l \epsilon |\beta_i|, \quad (8)$$

sujeito a:

$$\sum_{i=1}^l \beta_i = 0. \quad (9)$$

Assim, a função de regressão na Equação (1) pode ser reescrita da seguinte forma:

$$\mathbf{w} = \sum_{i=1}^l \beta_i \mathbf{x}_i \quad \text{e} \quad (10)$$

$$b = -\frac{1}{2}((\mathbf{w} \cdot \mathbf{x}_r) + (\mathbf{w} \cdot \mathbf{x}_s)). \quad (11)$$

A seguir, apresenta-se método SVR proposto neste trabalho, o qual baseia-se em uma adaptação de uma abordagem multitarefa para a aplicação da técnica de transferência de aprendizagem por parâmetros.

⁵Do original, em inglês, *Mean Squared Error*.

3.3. Técnica SVR adaptada para transferência de aprendizagem (ATL)

No estudo pioneiro desenvolvido em [Evgeniou and Pontil 2004] foi proposta uma abordagem de aprendizagem multitarefa para o problema de otimização SVM com base em grafos. Nesse caso, cada tarefa corresponde a um nó do grafo e a semelhança entre estas é codificada pelos pesos das arestas, por sua vez armazenados em uma matriz de adjacência A , em que $A_{st} \geq 0$.

Além disso, um termo regularizador $J(\mathbf{w}_1, \dots, \mathbf{w}_T)$ foi adicionado ao problema de otimização primal formulado na Equação (2):

$$J(\mathbf{w}_1, \dots, \mathbf{w}_t) = \frac{1}{4} \sum_{s=1}^T \sum_{t=1}^T A_{st} \|\mathbf{w}_s - \mathbf{w}_t\|^2, \quad (12)$$

em que \mathbf{w}_s e \mathbf{w}_t são os vetores de pesos associados a cada tarefa específica para a abordagem multitarefa.

O termo regularizador tem o papel de tornar os T modelos, obtidos pelas T tarefas, mais semelhantes entre si. Conseqüentemente, um modelo com um bom poder de generalização, obtido a partir de um conjunto de treinamento com um grande número de instâncias disponíveis, pode melhorar o desempenho de modelos ineficientes, construídos a partir de poucos dados.

No presente trabalho, propõe-se a formulação de um problema de otimização primal para transferência de aprendizagem via SVR, baseando-se na combinação das Equações (2) e (12), o que resulta em:

$$\min_{\mathbf{w}_s, \mathbf{w}_t, \xi_s^*, \xi_t^*, \xi_s, \xi_t} = \frac{1}{2}(\mathbf{w}_s^T \mathbf{w}_s) + \frac{1}{2}(\mathbf{w}_t^T \mathbf{w}_t) + J(\mathbf{w}_s, \mathbf{w}_t) + \frac{c}{2} \left(\sum_{i=1}^l (\xi_{si})^2 + \sum_{i=1}^l (\xi_{ti}^*)^2 \right) + \frac{c}{2} \left(\sum_{i=1}^l (\xi_{ti})^2 + \sum_{i=1}^l (\xi_{si}^*)^2 \right). \quad (13)$$

A esse modelo deu-se o nome de Regressão por Vetores de Suporte Adaptado para Transferência de Aprendizagem (ATL), o qual considera apenas duas tarefas, com a notação s para a tarefa de origem e t para a tarefa de destino.

Na Equação (13), os termos $\|\mathbf{w}_s\|^2$, $\|\mathbf{w}_t\|^2$ controlam a complexidade do modelo da tarefa específica, como acontece para o método SVR padrão. A função $J(\mathbf{w}_s, \mathbf{w}_t)$ representa o termo de regularização adicional que aumenta a semelhança dos vetores de pesos das tarefas de origem e destino. Assim, a transferência de aprendizagem é alcançada ao impor que a função f_t seja mais semelhante a função f_s . Uma ilustração de como a técnica transfere o conhecimento entre as tarefas é apresentado na Figura 4.

O termo regularizador $J(\mathbf{w}_s, \mathbf{w}_t)$ torna o modelo da tarefa de destino (\mathbf{w}_t) mais semelhante ao modelo da tarefa de origem (\mathbf{w}_s). Com isso, um modelo \mathbf{w}_t que é mais semelhante ao modelo \mathbf{w}_s possui melhor poder de predição para dados desconhecidos, o que resulta em uma melhor generalização do modelo.

Como na Equação (6), utilizando a função de perda ϵ -quadrática, a formulação dual assume a seguinte configuração:

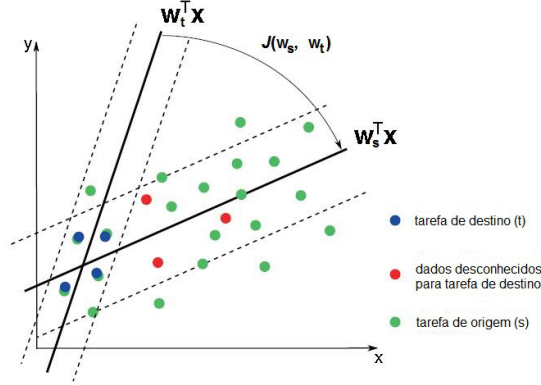


Figura 4. Ilustração do conhecimento transferido a partir da tarefa de origem, que contém uma grande quantidade de dados de treinamento (verde) para uma tarefa de destino, que contém poucos dados de treinamento (azul). Os intervalos de erro esperado ao redor das funções de regressão $w_s^T x$ e $w_t^T x$ são descritos pelas linhas tracejadas. Figura adaptada de [Rosenbaum et al. 2013].

$$\begin{aligned}
 \min_{\mathbf{w}_s, \mathbf{w}_t, \xi_s^*, \xi_t^*, \xi_s, \xi_t} &= \frac{1}{2}(\mathbf{w}_s^T \mathbf{w}_s) + \frac{1}{2}(\mathbf{w}_t^T \mathbf{w}_t) + \frac{1}{4}(\mathbf{w}_s^T - \mathbf{w}_t^T)(\mathbf{w}_s - \mathbf{w}_t) + \frac{c}{2} \left(\sum_{i=1}^l (\xi_{si})^2 + \sum_{i=1}^l (\xi_{si}^*)^2 \right) \\
 &+ \frac{c}{2} \left(\sum_{i=1}^l (\xi_{ti})^2 + \sum_{i=1}^l (\xi_{ti}^*)^2 \right) + \sum_{i=1}^l \alpha_{si} (y_{si} - \mathbf{w}_s^T \mathbf{x}_{si} - \epsilon_s - \xi_{si}) + \sum_{i=1}^l \alpha_{ti} (y_{ti} - \mathbf{w}_t^T \mathbf{x}_{ti} - \epsilon_t - \xi_{ti}) + \\
 &\sum_{i=1}^l \alpha_{si}^* (\mathbf{w}_s^T \mathbf{x}_{si} - y_{si} - \epsilon_s - \xi_{si}^*) + \sum_{i=1}^l \alpha_{ti}^* (\mathbf{w}_t^T \mathbf{x}_{ti} - y_{ti} - \epsilon_t - \xi_{ti}^*).
 \end{aligned} \tag{14}$$

Fazendo uma substituição, tal como na Equação (7), e calculando a derivada em relação a \mathbf{w}_s , \mathbf{w}_t , ξ_{si}^* , ξ_{ti}^* , ξ_{si} , ξ_{ti} , obtém-se a formulação dual do primal apresentado na Equação (13):

$$\begin{aligned}
 \min_{\beta_s, \beta_t} &= -\frac{3}{8} \sum_{i=1}^l \beta_{si} \beta_{sj} (\mathbf{x}_{si} \cdot \mathbf{x}_{sj}) - \frac{1}{4} \sum_{i=1}^l \beta_{si} \beta_{tj} (\mathbf{x}_{si} \cdot \mathbf{x}_{tj}) - \frac{3}{8} \sum_{i=1}^l \beta_{ti} \beta_{tj} \\
 &(\mathbf{x}_{ti} \cdot \mathbf{x}_{tj}) - \sum_{i=1}^l \frac{\beta_{si}^2}{2C} - \sum_{i=1}^l \frac{\beta_{ti}^2}{2C} + \sum_{i=1}^l \beta_{si} y_{si} - \sum_{i=1}^l |\beta_{si}| \epsilon_s + \sum_{i=1}^l \beta_{ti} y_{ti} - \sum_{i=1}^l |\beta_{ti}| \epsilon_t.
 \end{aligned} \tag{15}$$

4. Análise experimental

Atualmente, devido a grande quantidade de dados produzidos em todas as áreas da ciência e da indústria, existem muitas bases de dados relacionadas que permitem a aplicação das técnicas de transferência de aprendizagem. Diante deste fato, os cinco pares de conjuntos de dados utilizados (conjuntos de origem e de destino) são apenas uma seleção pequena de uma gama mais ampla de possibilidades. Além disso, assume-se que os conjuntos de dados são suficientemente relacionados para que a aplicação da técnica de transferência de aprendizagem seja viável.

O desempenho do método proposto neste trabalho foi avaliado em três conjuntos de dados de *benchmark* [Garcke and Vanck 2014] e dois conjuntos de dados químicos

[Girschick et al. 2012]. Para o conjunto de dados *Kin-family*⁶, definiu-se o subconjunto com instâncias rotuladas com *nm* (variância média não linear) como conjunto de origem e o subconjunto com instâncias rotuladas com *nh* (variância alta não linear) como conjunto de destino. O conjunto de dados *Earthquake* [Garcke and Vanck 2014] contém observações sobre terremotos ocorridos em diferentes países. Nesse caso, um modelo gerado a partir dos dados medidos na Califórnia, por exemplo, é adequado somente para fazer previsões futuras de terremoto para essa localidade, podendo não ser apropriado para previsões em outros países. No entanto, se os dados coletados no Japão forem muito escassos, um modelo gerado a partir de tais dados pode não apresentar um bom poder de generalização. Com isso, embora as distribuições de probabilidade dos dados da Califórnia e do Japão diferirem, em geral, é razoável supor que é possível utilizar o modelo gerado a partir dos dados da Califórnia para aprimorar o modelo gerado a partir dos dados coletados no Japão. Portanto, resolveu-se utilizar os dados coletados na Califórnia como conjunto de origem e os dados coletados no Japão como conjunto de destino.

O terceiro par de conjuntos de dados descreve os detalhes de chegada e partida para todos os voos⁷ comerciais nos Estados Unidos, contendo registros dos voos de outubro de 1987 a abril de 2008. Nesse caso, o valor a ser predito é o atraso de um determinado voo. Para os experimentos deste trabalho, definiu-se que os dados de 2007 seriam utilizados como conjunto de origem e os dados de 2008 como conjunto de destino.

Por fim, foram considerados conjuntos de dados químicos. O objetivo principal das pesquisas realizadas em química medicinal é prever o valor de inibição apresentado pelos compostos químicos em relação a um alvo biológico. Por exemplo, a medida de *IC50* indica quanto de uma determinada substância é necessário para bloquear a atividade de uma proteína ou enzima relacionada a uma doença ou alvo biológico. Assim, o objetivo será prever corretamente o valor de *IC50* nos conjuntos analisados. Para o quarto par de conjunto de dados, têm-se: DHFR-4q [Sutherland et al. 2003] como conjunto de origem e DHFR-S [Sutherland et al. 2004] como conjunto de destino, sendo DHFR⁸ uma enzima que reduz a dihidrofolato. O quinto par de conjuntos de dados é formado por: COX2-4q [Sutherland et al. 2003] (conjunto de origem) e COX2-S [Sutherland et al. 2004] (conjunto de destino), representando moléculas candidatas a fármacos que tem como alvo diretamente o COX2⁹, uma enzima responsável pela inflamação e dor.

4.1. Discussão dos resultados

[Rosenbaum et al. 2013] afirmam que a transferência de aprendizagem geralmente é benéfica se aplicada a partir de um conjunto de dados com maior quantidade de instâncias disponíveis para um conjunto de dados com pouca quantidade de instâncias disponíveis. Para analisar em quais situações a abordagem de transferência de aprendizagem agrega benefícios aos modelos de destino, os experimentos foram repetidos considerando diferentes tamanhos de conjuntos de destino, como variação de seus tamanhos originais.

Os conjuntos de dados de destino descritos acima foram testados individualmente com o método SVR. Além disso, nos experimentos, utilizou-se a técnica de amostragem

⁶Conjuntos de dados *Kin-family* disponível no repositório delve: <http://www.cs.toronto.edu/~delve/data/kin/desc.html>.

⁷Conjunto de dados disponível em: <http://stat-computing.org/dataexpo/2009/>.

⁸Do original, em inglês, *Dihydrofolate Reductase*.

⁹Do original, em inglês, *Cyclooxygenase-2*.

validação cruzada k -fold, com $k = 5$ e como medida de desempenho o erro quadrático médio (MSE). Para a execução do método SVR, foi utilizada a biblioteca *e1071* do software R¹⁰ com os parâmetros: *SVM-Type = eps-regression*, *SVM-Kernel = linear*, *cost=100*, *10 e 0.1*).

Para a execução do método SVR-ATL, é necessário gerar os modelos de origem e de destino. Para o modelo de destino considerou-se os conjuntos de destino gerados para o método SVR e para o modelo de origem utilizou-se os conjuntos de origem referenciados acima com tamanho total (sem remoção de instâncias). Os modelos foram avaliados considerando os dados desconhecidos da tarefa de destino. Para a configuração do método, empregou-se a função *quadprog*¹¹ do Matlab. Para a execução do problema de otimização dual [ver Equação (15)] via função *quadprog*, passou-se por parâmetro a matriz apresentada a seguir:

$$\begin{bmatrix} -3/4 * X_s * X_s' - eye(N) * 1/C & -1/4 * X_s * X_t' \\ -1/4 * X_t * X_s' & -3/4 * X_t * X_t' - eye(M) * 1/C \end{bmatrix}. \quad (16)$$

Alem disso, utilizamos *Kernel* linear e variamos também o valor do parâmetro $C(100, 10 e 0.1)$ da função. Os resultados obtidos nos experimentos são apresentados na Tabela 1, em que observa-se o *trade-off* entre o desempenho do método SVR com a abordagem proposta e a quantidade de dados de treinamento no conjunto de dados de destino. Para isso, considerou-se conjuntos de destino com diferentes tamanhos (valores da variável n , apresentados na tabela).

Os melhores resultados (menores valores de EQM) foram destacados e mostram que o método SVR-ATL obteve melhores resultados em todos os cenários (diferentes valores de n) para os conjuntos de destino Voo 2008, *Earthquake* Japão, DHFR-S e COX2-S. A abordagem apenas não apresentou melhoria para a tarefa de destino nos experimentos com o conjunto de dados *kin-family nh*. Isso pode ser decorrente de uma possível falta de relação entre os dados de origem e os dados de destino.

Os resultados apresentados comprovam que a abordagem de transferência de aprendizagem pode ser uma ótima alternativa para melhorar a acurácia do modelo gerado para a tarefa de destino quando este não possui dados suficientes para gerar um modelo com um bom poder de generalização.

5. Considerações finais

Este trabalho apresentou o método SVR adaptado para transferência de aprendizagem. Tal abordagem visa a melhorar o poder de generalização de um modelo gerado a partir dos dados de treinamento de uma tarefa de destino, quando estes não são suficientes. Os resultados experimentais mostraram que a abordagem proposta foi superior ao método SVR tradicional. Em trabalhos futuros, pretende-se comparar o desempenho do método SRV ATL com técnicas de transferência de aprendizagem por instâncias, como o método TrAdaBoost [Pardoe and Stone 2010].

6. Agradecimentos

Gostaríamos de agradecer ao Grupo de Pesquisa de Inteligência Artificial (GRIA) pela troca de conhecimento e a Coordenação de Aperfeiçoamento de Pessoal de Nível Su-

¹⁰<https://www.r-project.org/>.

¹¹<https://www.mathworks.com/help/optim/ug/quadprog.html>.

Tabela 1. Resultados médios obtidos pelos métodos SVR e SVR-ATL com validação cruzada 5-fold e considerando a medida EQM para a tarefa de destino.

Conjunto de destino	n	SVR			SVR-ATL		
		<i>cost</i> =0.1	<i>cost</i> =10	<i>cost</i> =100	<i>C</i> =0.1	<i>C</i> =10	<i>C</i> =100
Voo 2008	50	5,14E-02	5,14E-02	5,14E-02	1,20E-02	1,23E-06	2,40E-08
	200	4,96E-03	4,96E-03	4,96E-03	1,84E-04	1,89E-08	2,49E-09
	400	5,82E-02	5,82E-02	5,82E-02	8,75E-03	1,06E-06	1,32E-08
	800	2,97E-03	2,97E-03	2,97E-03	2,89E-03	3,20E-07	4,98E-09
<i>Kin-family</i> nh	50	5,90E-02	6,55E-02	6,58E-02	4,61E-01	4,53E-01	4,53E-01
	100	4,64E-02	4,66E-02	4,65E-02	5,46E-01	5,46E-01	5,46E-01
	150	5,84E-02	5,88E-02	5,88E-02	5,14E-01	5,14E-01	5,14E-01
	200	4,88E-02	4,95E-02	4,95E-02	5,50E-01	5,51E-01	5,51E-01
<i>Earthquake</i> Japão	20	1,54E+00	9,05E-01	1,01E+01	1,36E+00	8,86E-01	8,86E-01
	30	1,52E+00	1,01E+00	1,08E+01	1,13E+00	7,74E-01	7,77E-01
	50	7,00E-01	7,35E-01	5,13E+00	7,17E-01	5,47E-01	5,52E-01
	70	7,99E-01	9,35E-01	4,87E+00	7,09E-01	7,34E-01	7,59E-01
DHFR-S	97	1,35E+00	1,36E+00	1,42E+00	1,27E+00	1,19E+00	1,19E+00
	195	1,47E+00	1,45E+00	1,45E+00	1,38E+00	1,37E+00	1,37E+00
	293	1,29E+00	1,28E+00	1,29E+00	1,29E+00	1,28E+00	1,28E+00
	391	1,22E+00	1,26E+00	1,26E+00	1,22E+00	1,23E+00	1,23E+00
COX2-S	79	1,44E+00	1,79E+00	1,82E+00	1,41E+00	1,44E+00	1,44E+00
	158	1,96E+00	2,03E+00	2,02E+00	1,76E+00	1,78E+00	1,78E+00
	237	1,80E+00	1,83E+00	1,83E+00	1,68E+00	1,69E+00	1,69E+00
	317	1,87E+00	1,88E+00	1,89E+00	1,75E+00	1,74E+00	1,74E+00

perior (CAPES)¹², a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP)¹³ e a *International Business Machines* (IBM)¹⁴ pelo suporte financeiro.

Referências

- Bickel, S., Sawade, C., and Scheffer, T. (2009). Transfer learning by distribution matching for targeted advertising. In *Advances in neural information processing systems*, pages 145–152.
- Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.
- Cristianini, N. and Shawe-Taylor, J. (2000). An introduction to support vector machines.
- Evgeniou, T. and Pontil, M. (2004). Regularized multi-task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM.
- Fitzgerald, T. and Thomaz, A. L. (2015). Skill demonstration transfer for learning from demonstration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, pages 187–188. ACM.

¹²<https://http://www.capes.gov.br/>.

¹³<http://www.fapesp.br/>

¹⁴<https://www.ibm.com/br-pt/>

- Garcke, J. and Vanck, T. (2014). Importance weighted inductive transfer learning for regression. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 466–481. Springer.
- Girschick, T., Rückert, U., and Kramer, S. (2012). Adapted transfer of distance measures for quantitative structure-activity relationships and data-driven selection of source datasets. *The Computer Journal*, page bxs092.
- Gunn, S. R. et al. (1998). Support vector machines for classification and regression. *ISIS technical report*, 14.
- Lima, C. A. d. M. (2004). *Comitê de Máquinas: uma abordagem unificada empregando máquinas de vetores-suporte*. PhD thesis, Universidade Estadual de Campinas.
- Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 22(10):1345–1359.
- Pardoe, D. and Stone, P. (2010). Boosting for regression transfer. In *Proceedings of the 27th international conference on Machine learning (ICML-10)*, pages 863–870.
- Rosenbaum, L., Dörr, A., Bauer, M. R., Boeckler, F. M., and Zell, A. (2013). Inferring multi-target qsar models with taxonomy-based multi-task learning. *J. Cheminformatics*, 5(1-2):33.
- Satpal, S. and Sarawagi, S. (2007). Domain adaptation of conditional probability models via feature subsetting. In *Knowledge Discovery in Databases: PKDD 2007*, pages 224–235. Springer.
- Smola, A. J. et al. (1996). Regression estimation with support vector learning machines. *Master's thesis, Technische Universit at M unchen*.
- Sutherland, J. J., O'brien, L. A., and Weaver, D. F. (2003). Spline-fitting with a genetic algorithm: A method for developing classification structure-activity relationships. *Journal of chemical information and computer sciences*, 43(6):1906–1915.
- Sutherland, J. J., O'Brien, L. A., and Weaver, D. F. (2004). A comparison of methods for modeling quantitative structure-activity relationships. *Journal of medicinal chemistry*, 47(22):5541–5554.
- Tan, B., Song, Y., Zhong, E., and Yang, Q. (2015). Transitive transfer learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1155–1164. ACM.
- Wang, S. and Li, Z. (2014). A new transfer learning boosting approach based on distribution measure with an application on facial expression recognition. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 432–439. IEEE.

Sistema *Fuzzy* Genético Baseado em Regras: Um Operador de Pré-Seleção com Foco na Diversidade das Soluções

Eduardo Fernando Velludo Prado¹, Maykon Rocha Santana², Tiago Pinho da Silva³, Marcela Aparecida Aniceto dos Santos⁴, Heloisa de Arruda Camargo⁵

¹²³⁴⁵Departamento de Computação

Universidade Federal de São Carlos (UFSCar) – São Carlos – SP – Brasil

{eduardovelludo¹, maykonsantana², marcelaaniceto⁴}@gmail.com,
{tiagopinholds³, heloisa⁵}@dc.ufscar.br

Abstract. *The parent's selection in the (2+2)M-PAES does not consider the disposition of the solutions in the Pareto Front which can result in the selection of solutions near each other, increasing the convergence's probability to a local optimum. This paper proposes a pre-selection operator for (2+2)M-PAES focusing on the solution diversity to improve genetic variability and performance. Experiments with the proposed approach resulted in final solutions with better average accuracy when compared to other approach reported in the literature.*

Resumo. *A seleção dos pais no algoritmo (2+2)M-PAES não considera a disposição das soluções na Fronteira de Pareto. Esse fato pode fazer com que essas soluções advenham de regiões próximas o que aumenta a chance de convergência para ótimos locais. O objetivo é desenvolver um operador de pré-seleção que atue com foco na diversidade das soluções proporcionando uma maior variabilidade genética e aumentando a chance de obtenção de melhores soluções. A abordagem proposta foi desenvolvida por meio da adição do operador de pré-seleção ao algoritmo (2+2)M-PAES. Testes na abordagem proposta e em uma abordagem da literatura demonstraram que o algoritmo proposto obteve um aumento na média da acurácia das soluções encontradas.*

1. Introdução

Segundo Kim, Oh e Pedrycz (2017) Sistemas Baseados em Regras são comumente encontrados na construção de modelos de inteligência artificial, particularmente em modelos *Fuzzy*. A união entre os Sistemas Baseados em Regras com os modelos *Fuzzy* formam os Sistemas *Fuzzy* Baseados em Regras (SFBR). Os SFBR são tipos de Sistemas *Fuzzy* (SF) usados com sucesso em aplicações nas áreas de controle, classificação, mineração de dados, etc. Essas aplicações podem ser vistas, dentre outros, nos trabalhos de Caridá, Morandin e Tuma (2015), Riid e Preden (2017) e Riza et al. (2017).

Os Algoritmos Genéticos (AG) em conjunto com os SFBR formam os Sistemas *Fuzzy* Genéticos Baseados em Regras (SFGBR). As aplicações dos AG nos SFBR geraram bons resultados que podem ser observados nos trabalhos de Córdón et al. (2004) e Abadeh, Mohamadi e Habibi (2011).

Todavia, os AG foram inicialmente concebidos para explorar o espaço de soluções na busca por apenas uma solução que fosse a melhor para resolver um dado problema. Especificamente nos SFGBR considerava-se, a princípio, a busca por soluções que apresentavam resultados voltados somente para o objetivo relacionado à acurácia. Segundo Fazzolari et al. (2013) objetivos relacionados à interpretabilidade do sistema foram negligenciados até meados da década de 1990. Essa interpretabilidade é definida, segundo Gacto, Alcalá e Herrera (2011) como a capacidade de um sistema em expressar o seu comportamento de uma forma compreensível, sendo esse, um objetivo que depende da pessoa que faz a avaliação.

Para suprir as carências dos AG quando da aplicação em problemas que exigem mais de um objetivo, surgiram então os Algoritmos Genéticos Multiobjetivo (AGMO) e a junção desses com os Sistemas *Fuzzy* formaram os chamados Sistemas *Fuzzy* Genéticos Multiobjetivo (SFGMO).

Geralmente, em SFGMO considera-se, para o processo de otimização, os objetivos de acurácia e interpretabilidade. Esses objetivos são conflitantes, ou seja, o aumento de um leva à diminuição do outro. Nesse caso, é obtido, não apenas uma, mas um conjunto de soluções com esses objetivos (acurácia e interpretabilidade) balanceados entre si.

Uma forma possível de lidar com objetivos conflitantes é com o uso da Técnica de Pareto. Essa técnica foi criada pelo economista Franco-Italiano Vilfredo Pareto em 1896 e possibilita, no contexto dos SFGMO, a otimização em conjunto dos objetivos acurácia e interpretabilidade. Essa otimização é realizada por intermédio da ponderação entre os objetivos, encontrando um conjunto de soluções. As melhores soluções ficam dispostas na chamada Fronteira de Pareto.

Em SFGMO o algoritmo (2+2)M-PAES¹ pode ser usado na resolução de problemas de otimização que envolvem os objetivos conflitantes de acurácia e interpretabilidade. Esse algoritmo seleciona as soluções que serão usadas para obtenção dos indivíduos da nova geração de forma aleatória, sem considerar, no entanto, qual a disposição dessas soluções na Fronteira de Pareto. Esse fato pode fazer com que as soluções selecionadas advenham de locais da Fronteira de Pareto que estejam próximos uns dos outros o que pode diminuir a variedade genética e, por consequência, aumentar a probabilidade da ocorrência de convergência para ótimos locais.

Tendo isso em vista, o objetivo nesse trabalho é o desenvolvimento de um operador de pré-seleção que atue com foco na diversidade das soluções. A ideia é proporcionar uma maior variedade genética o que pode levar à fuga de ótimos locais e, por conseguinte, resultar na exploração de regiões do espaço de busca que podem levar à obtenção de melhores soluções.

O artigo está organizado em seis seções. Na seção II o algoritmo PAES e suas variações será apresentado. Na seção III serão apresentadas as questões referente à dispersão de fronteira em AGMO. Na seção IV a proposta será apresentada. Na seção V tem-se os resultados e as discursões a respeito dos mesmos e na seção VI as conclusões a respeito do trabalho realizado.

¹ Mais detalhes sobre o (2+2)M-PAES pode ser visto na seção 2.1

2. Pareto Archived Evolution Strategy

O algoritmo PAES (*Pareto Archived Evolution Strategy*) foi apresentado no trabalho de Knowles e Corne (1999) com o objetivo de possibilitar a geração de diversas soluções no conjunto de soluções ótimas em uma Fronteira de Pareto. O algoritmo foi desenvolvido, segundo Knowles e Corne (1999), com intuito de alcançar 2 objetivos principais. O primeiro relacionado a área de atuação do algoritmo durante uma busca por soluções. A ideia foi a de confinar a busca em uma área local devendo-se usar apenas o operador de mutação que provocasse uma pequena mudança no gene de modo a mover a busca da solução corrente para a solução vizinha. Essa característica, tornou-o diferente dos Algoritmos Genéticos Multiobjetivo da época como o *Niched Pareto Genetic Algorithm* e o *Nondominated Sorting Genetic Algorithm* (NSGA). O segundo objetivo estava relacionado ao tratamento das soluções não-dominadas. Neste caso, todas as soluções não-dominadas possuem o mesmo peso na busca pela fronteira ideal. Um fluxograma do PAES é apresentado na Figura 1.

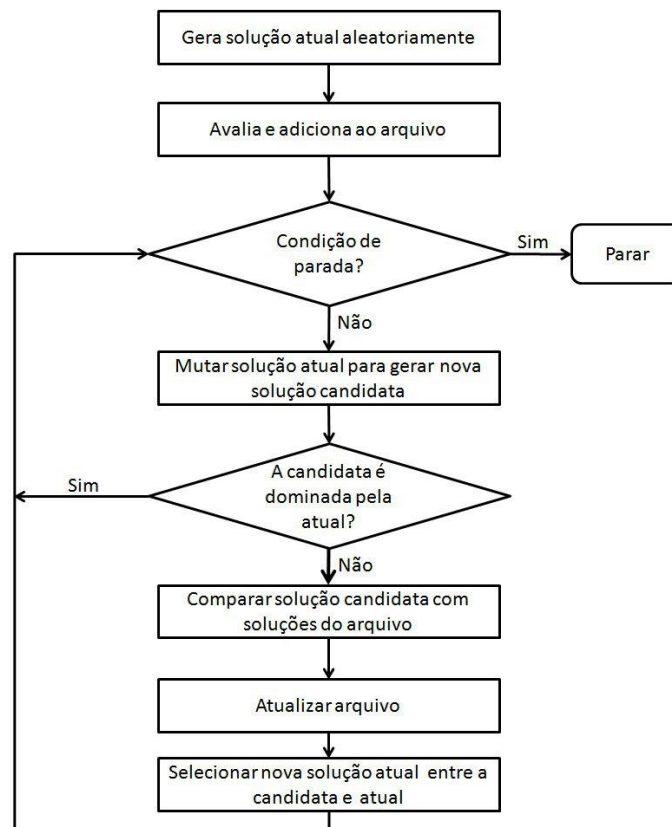


Figura 1 - Fluxograma do Algoritmo PAES adaptado de Knowles e Corne (1999)

Nesse fluxograma, o PAES, inicialmente, gera uma solução randômica chamada de Solução Corrente. A seguir, essa Solução Corrente é avaliada e adicionada ao arquivo. A Solução Corrente sofre mutação a fim de para gerar uma nova solução chamada de Solução Candidata. A Solução Candidata é avaliada e, posteriormente, verifica-se a dominância em relação à Solução Corrente. A partir desse ponto a Solução Candidata é ou não adicionada ao arquivo a depender de critérios de dominância de soluções abordadas nos artigos de Knowles e Corne (1999) e Knowles e Corne (2000). A cada iteração uma solução candidata é gerada e o PAES continua a busca por uma Fronteira de Pareto otimizada para determinado problema.

A seguir uma variante do PAES chamada de 2+2 M-PAES é apresentada sendo essa a usada na abordagem proposta no presente artigo.

2.1. (2+2)M-PAES

O (2+2)M-PAES foi apresentado no artigo de Cococcioni et al. (2007) como uma variação do PAES. As mudanças visam proporcionar a geração de Bases de Regras *Fuzzy* em Sistemas *Fuzzy* Baseados em Regras.

As principais mudanças estão na adição do operador de cruzamento, na adição de um novo operador de mutação e da utilização de 2 Soluções Correntes na geração de 2 novas Soluções Candidatas. O operador de cruzamento é implementado usando 1 ponto de corte para geração das novas soluções. Esse ponto de corte é escolhido de modo a não dividir a regra *Fuzzy*, ou seja, o ponto de corte é definido aleatoriamente ao fim de uma das regras intermediárias contidas no cromossomo.

O (2+2)M-PAES possui 2 operadores de mutação. O primeiro é responsável pela criação de novas regras *Fuzzy* e o segundo é responsável pela mutação de alguns dos genes dos cromossomos dos indivíduos. Esses genes representam as regras *Fuzzy* e essas mutações afetam diretamente os antecedentes e/ou os consequentes dessas regras.

O funcionamento do algoritmo pode ser visto na Figura 2.

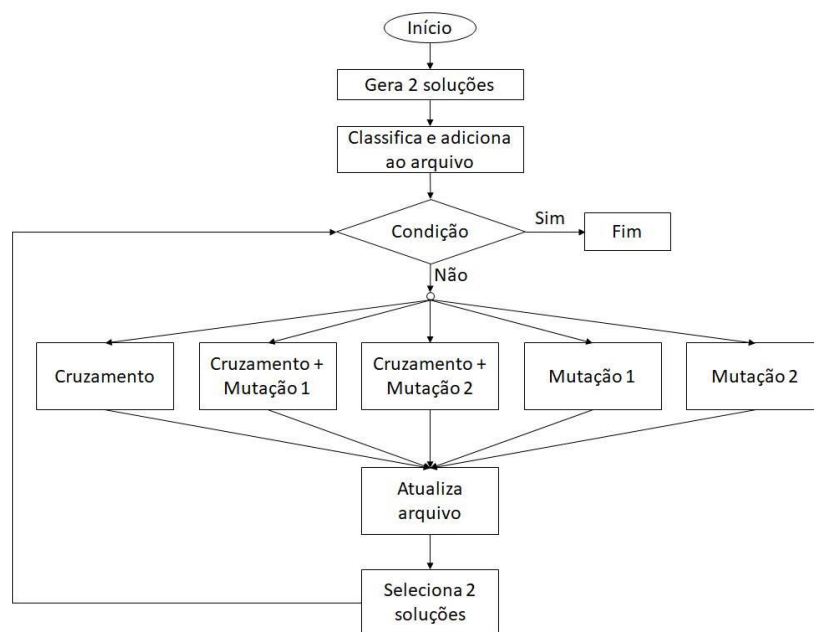


Figura 2 - Fluxograma do Algoritmo (2+2)M-PAES

No início são geradas 2 soluções chamadas de Soluções Correntes. Essas soluções são adicionadas ao arquivo que armazena as soluções não-dominadas. Em seguida, uma dentre 5 possibilidades pode ocorrer. A primeira gera 2 Soluções Candidatas a partir apenas do operador de cruzamento. A segunda além do operador de cruzamento, usa o operador de Mutação 1 (responsável por criar regras *Fuzzy*). A terceira possibilidade diz respeito ao uso do operador de cruzamento em conjunto com o operador de mutação 2 (responsável por alterar os genes dos cromossomos). A quarta possibilidade está relacionada à geração de 2 soluções candidatas a partir somente do operador de mutação

1. E, por último, a quinta possibilidade está relacionada a utilização apenas do operador de mutação 2.

Vale ressaltar que, cada uma dessas possibilidades pode ocorrer de acordo com uma determinada probabilidade definida pelo programador a depender do problema abordado. A seguir, após a aplicação do(s) operador(es), o arquivo é atualizado e 2 novas Soluções Correntes são selecionadas aleatoriamente. O Critério de Parada é verificado e, caso necessário, uma nova iteração é realizada.

Na Figura 3 é apresentado um cromossomo do algoritmo (2+2)M-PAES.

No exemplo da Figura 3 tem-se o cromossomo representando uma solução de determinado problema (uma base de regras *Fuzzy*) com R_1 , R_2 e R_3 representando as regras *Fuzzy*, A_1 , A_2 e A_3 representando os antecedentes das regras e C_1 , C_2 e C_3 representando os consequentes.

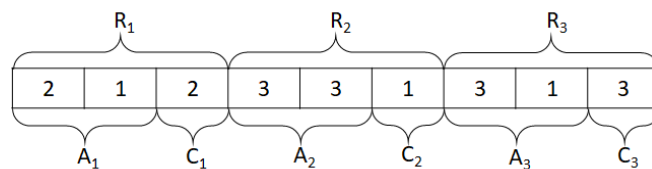


Figura 3 - Cromossomo do algoritmo (2+2)M-PAES

3. Dispersão de Fronteira em Algoritmos Genéticos Multiobjetivo

Em algoritmos genéticos multiobjetivo a ideia está no balanceamento entre os objetivos na busca por soluções de determinados problemas. Em algumas situações, as soluções encontram-se desbalanceadas, ou seja, o conjunto de soluções encontrado pode estar beneficiando algum dos objetivos. Isso ocorre quando, em determinadas situações, os operadores geram soluções que, no decorrer das iterações, beneficiam mais um objetivo do que o(s) outro(s). Existem, no entanto, mecanismos que buscam minimizar esse efeito tornando as soluções da fronteira gerada mais dispersas entre si.

Além disso, dispersar soluções na Fronteira de Pareto é uma forma de aumentar a variabilidade genética e, conseqüentemente, um método para explorar regiões diferentes do espaço de busca a fim de fugir de ótimos locais.

No algoritmo SPEA2, um dos métodos usados para aumentar a dispersão dos indivíduos ocorre no processo de exclusão de determinadas soluções. Nesse algoritmo, em determinadas situações, há a necessidade de excluir algumas das soluções encontradas. Nesse caso, o algoritmo prioriza a exclusão de soluções que estejam numa região da fronteira mais densa, ou seja, uma região que possua uma maior concentração de soluções e uma menor variedade genética. Esse fato corrobora com a diminuição da possibilidade de geração de fronteiras que possuam regiões com maior concentração de soluções do que outras.

O mesmo ocorre no algoritmo NSGA-II quando há a necessidade de selecionar indivíduos para a próxima iteração. Nessa situação, o operador de seleção privilegia as soluções que estejam em áreas menos densas aumentando a probabilidade da geração de fronteiras cujas soluções estejam mais dispersas e com maior variedade genética.

Uma versão do NSGA-II, chamada de NSGA-DO foi desenvolvida por Pimenta e Camargo (2015) com o intuito de realizar melhorias nas questões relacionadas à seleção

de indivíduos para a próxima geração. Essa melhora tem o enfoque voltado para o direcionamento da convergência no sentido dos chamados pontos ideais. Esses pontos são definidos de acordo com cálculos realizados para determinar os locais ideais das soluções para obtenção de uma Fronteira de Pareto perfeitamente distribuídas. Nesse caso, as soluções mais próximas dos pontos ideais serão as escolhidas para a obtenção dos indivíduos da próxima geração.

A dispersão no algoritmo PAES ocorre de forma similar ao algoritmo SPEA2. As soluções a serem descartadas são escolhidas de acordo com a densidade da fronteira atual. Quanto maior a concentração de soluções em determinada região da fronteira, maior será a probabilidade de uma dessas soluções não ser adicionada ao conjunto que conterá as soluções responsáveis por gerar os indivíduos da próxima geração.

Portanto, pode-se observar que, em geral, os AGMO implementam formas de dispersão a fim de obter fronteiras melhor balanceadas no que diz respeito à distância entre as soluções encontradas. Estudos como o do NSGA-DO demonstram que as questões relacionadas à dispersão em AGMO têm despertado interesse nos últimos anos.

4. Proposta

A proposta deste trabalho é o desenvolvimento de um operador de pré-seleção que atue com foco na diversidade das soluções. A ideia é proporcionar uma maior variabilidade genética o que pode levar à fuga de ótimos locais e, por conseguinte, resultar no aumento da probabilidade de obtenção de melhores soluções.

A abordagem proposta foi desenvolvida por intermédio da adição do Operador de Pré-Seleção ao algoritmo multiobjetivo (2+2)M-PAES. Nesse caso, o Operador de Pré-seleção fornece um conjunto de soluções selecionadas da Fronteira de Pareto atual. Esse conjunto de soluções é escolhido de forma alternada, ou seja, a cada solução adicionada ao conjunto de pré-seleção a solução imediatamente seguinte é descartada.

Esse fato diminui, mas não elimina, a probabilidade de que soluções próximas sejam selecionadas para serem os possíveis pais dos indivíduos da próxima geração.

No (2+2)M-PAES de Cococcioni et al. (2007) a seleção de pais é feita de forma aleatória, ou seja, dentre um conjunto de soluções presentes no arquivo selecionam-se 2 soluções aleatoriamente para gerar os indivíduos da próxima geração. Esse fato permite que soluções que estejam muito próximas umas das outras sejam selecionadas possibilitando a geração de indivíduos que levam a uma Fronteira de Pareto desbalanceada.

Na proposta do trabalho, mantém-se a aleatoriedade natural da seleção de pais do (2+2)M-PAES de Cococcioni et al. (2007) a fim de manter a estocasticidade do algoritmo. No entanto, há uma etapa de pré-seleção dos candidatos aos pais dos indivíduos da próxima geração. Essa pré-seleção é realizada levando em consideração as soluções vizinhas na fronteira. Portanto, soluções que estejam diretamente ao lado umas das outras não são pré-selecionadas (exceto em alguns casos no extremo da fronteira) o que evita que soluções vizinhas sejam usadas na geração de indivíduos da próxima geração. Esse fato corrobora com a geração de soluções que levam a uma fronteira mais dispersa (maior variedade genética e, conseqüentemente, maior exploração do espaço de busca) quando comparada com situações nas quais pontos vizinhos são usados para gerar novas soluções.

A seguir o algoritmo (2+2)M-PAES com a modificação proposta será apresentado.

4.1. (2+2)M-PAES Modificado

No algoritmo desenvolvido é implementado um método de pré-seleção de pais para geração de uma Fronteira de Pareto onde as soluções estejam devidamente dispersas.

Primeiramente é realizada a ordenação das soluções do arquivo de acordo com um dos objetivos. A seguir, as soluções ordenadas são selecionadas de forma alternada (a partir do primeiro indivíduo) para geração do conjunto de soluções candidatas a pais dos indivíduos da próxima geração. Vale ressaltar que a última solução sempre deve ser selecionada, pois a primeira e a última solução formam a extensão da Fronteira de Pareto atual. Esse método pode ser visto na Figura 4.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figura 4 - Exemplo de soluções candidatas a serem as possíveis progenitoras dos indivíduos da próxima geração (Pontos escuros)

Na Figura 4 os pontos escuros simbolizam as soluções escolhidas para fazerem parte do conjunto de soluções candidatas. Os outros pontos não são considerados para serem possíveis pais dos indivíduos da próxima geração. Entre essas soluções candidatas são sorteadas 2 soluções para serem os pais dos indivíduos da próxima geração.

O pseudocódigo do algoritmo da seleção de soluções pode ser visto a seguir.

Algorithm 1: Seleção de soluções proposta

Input: NumeroDeSoluções que o arquivo deve contém para se iniciar a seleção
Output: Lista com 2 soluções selecionadas na variável **Selecionados**

```

1 if TamanhoDaLista(Arquivo) > NumeroDeSoluções then
2   ListaOrdenada ← Ordenar(Arquivo);
3   ListaDeSorteio ← CriaLista();
4   ListaDeSorteio.Adicionar(UltimoElemento(ListaOrdenada));
5   Contador ← 0;
6   while Contador < TamanhoDaLista(ListaOrdenada) do
7     ListaDeSorteio.Adicionar(ListaOrdenada[Contador]);
8     Contador ← Contador + 2;
9   Selecionados ← SelecionarAleatoriamente(ListaDeSorteio);
10 else
11   Selecionados ← SelecionarAleatoriamente(Arquivo);

```

No algoritmo, a variável “NumeroDeSoluções” tem seu valor definido pelo programador. Esse valor indica a partir de quantas soluções presentes no arquivo a pré-seleção começa a ser executada. O algoritmo inicia verificando se o número de soluções presentes no arquivo é maior que o valor configurado de entrada. Se o número de soluções do arquivo for maior que a variável “NumeroDeSoluções” então as soluções do arquivo são ordenadas e inseridas em uma lista chamada de “ListaOrdenada” (linha 2). Na linha 4, o último elemento da lista ordenada é adicionado à uma nova lista chamada de “ListaDeSorteio” que será usada no sorteio para a escolha dos 2 novos pais. A linha 6 contém o laço responsável por adicionar à lista de sorteio as soluções da lista ordenada alternadamente. Após o preenchimento da lista de sorteio com soluções da lista ordenada

é realizado uma seleção aleatória na lista de sorteio que retorna e armazena em “Selecionados” as 2 soluções que serão usadas como pais da próxima geração.

Na Figura 5 é apresentado um gráfico com soluções formando a Fronteira de Pareto. Os pontos em “x” representam as soluções candidatas a pais dos indivíduos da próxima geração. Os pontos em “●” não serão considerados para fazer parte do conjunto de possíveis pais.

A seguir o cálculo da dispersão usado nesse trabalho será apresentado.

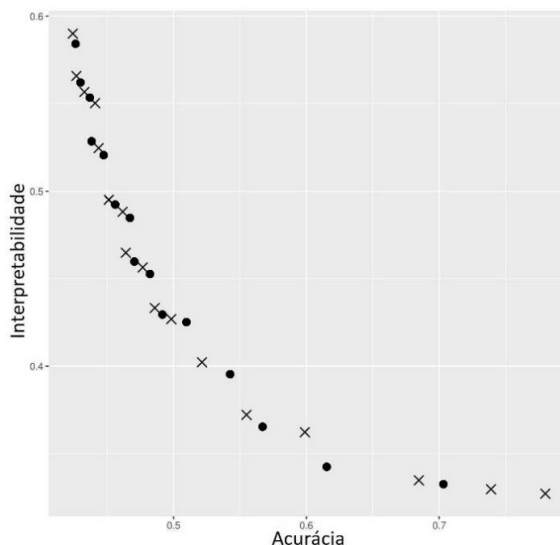


Figura 5 – Gráfico com soluções que forma a Fronteira de Pareto

4.2. Cálculo da Dispersão

Para analisar os resultados obtidos nos testes realizados, é proposto nesse artigo uma medida de dispersão de fronteira que se baseia no artigo de Deb et al. (2002).

Em Deb et al. (2002) uma medida de dispersão foi proposta para o cálculo de dispersão de soluções do NSGA-II e leva em consideração o comprimento da fronteira e a posição dos pontos da fronteira em relação às posições consideradas ideais. A fórmula usada em Deb et al. (2002) é:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}} \quad (1)$$

Onde Δ é medida usada para definir o quão dispersa estão as soluções da Fronteira de Pareto. Quanto menor o Δ melhor a dispersão das soluções na fronteira sendo 0 (zero) a dispersão perfeita. É importante ressaltar que Deb et al. (2002) afirmam que o pior caso de dispersão não necessariamente resulta em um Δ igual a 1.

As variáveis d_f e d_l são as distâncias euclidianas entre as soluções extremas da fronteira ideal e as soluções da borda do conjunto não dominado obtido. A variável \bar{d} representa a média de todas as distâncias entre as soluções da fronteira obtida. A variável d_i representa as distâncias entre as soluções i e $i+1$ da fronteira obtida. A variável N representa o número de soluções presentes na fronteira obtida.

Nessa medida considera-se que, no cálculo da dispersão de soluções do NSGA-II, sabe-se previamente qual é a fronteira ideal. Nessa situação a fórmula proposta contém as variáveis d_f e d_l usadas para definir o deslocamento da fronteira ideal em relação à fronteira obtida como pode ser visto na Figura 6.

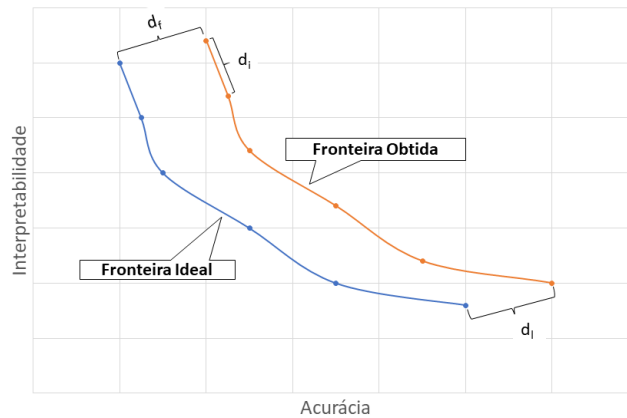


Figura 6 - Fronteira Obtida e Fronteira Ideal para o NSGA-II adaptado de Deb et al. (2002)

No cálculo proposto no presente artigo, considera-se que a fronteira ideal está na mesma posição da fronteira obtida e, portanto, os valores de d_f e d_l são iguais a zero. Desse modo, a fórmula proposta para o cálculo da dispersão no (2+2)M-PAES modificado é:

$$\Delta = \frac{\sum_{i=1}^{N-1} |d_i - \bar{d}|}{(N-1)\bar{d}} \quad (2)$$

Vale ressaltar que essa medida de dispersão não é usada no algoritmo e sim como medida para os testes durante a fase de experimentação.

5. Resultados

Para permitir testar o desempenho da metodologia proposta, experimentos foram realizados envolvendo a base de dados *Box-Jenkins Gas Furnace* (BJGF). Essa base é obtida por intermédio da combinação em sistema de aquecimento de ar e gás metano para formar uma mistura de gás contendo CO_2 . Essa base foi usada no artigo de Cococcioni et al. (2007) e os resultados obtidos nesse artigo foram usados como critério de comparação com a abordagem aqui proposta.

Os valores de cruzamento e mutação foram os mesmos utilizados por Cococcioni et al. (2007) e os experimentos foram executados 500 vezes em cada algoritmo (cada execução realizou 200000 iterações). Cada execução gera, no máximo, 100 Bases de Regras (tamanho máximo do arquivo no (2+2)M-PAES Modificado) e cada Base de Regras gerada possui uma acurácia e uma interpretabilidade correspondente. A Tabela 1 mostra os resultados obtidos em relação à acurácia.

A tabela é dividida em treino e teste tanto para o artigo de Cococcioni et al. (2007) quanto para a abordagem proposta. O Erro Mínimo possui o melhor valor de acurácia que uma base obteve e o Erro Máximo o pior. O Erro Médio é a somatória do erro de todas as bases que foram obtidas em todas as execuções dividido pelo número de bases.

Na Tabela 1 (referente à acurácia) observa-se que o resultado obtido para o Erro Mínimo na fase de Treino foi 0.0195 menor na abordagem de Cococcioni et al. (2007) em relação a abordagem proposta. Na fase de Teste a abordagem de Cococcioni et al. (2007) foi 0.0533 maior em relação a abordagem proposta.

Na mesma tabela observa-se que o resultado obtido para o Erro Máximo na fase de Treino foi 8.5663 maior na abordagem de Cococcioni et al. (2007) em relação a abordagem proposta. O mesmo ocorreu na fase de Teste onde a abordagem Cococcioni et al. (2007) foi 11.1543 unidades maior em relação a abordagem proposta.

Ainda na Tabela 1 o resultado obtido para o Erro Médio na fase de Treino foi 0.0119 menor na abordagem de Cococcioni et al. (2007) em relação a abordagem proposta. Já na fase de Teste a abordagem de Cococcioni et al. (2007) foi 0.2472 maior em relação a abordagem proposta.

Obteve-se para o Desvio Padrão na Tabela 1 na fase de Treino um resultado 0.0859 maior na abordagem de Cococcioni et al. (2007) em relação a abordagem proposta. Já na fase de Teste o Desvio Padrão da abordagem de Cococcioni foi 0.2152 maior em relação a abordagem proposta.

Esses resultados demonstram que nos testes, com a abordagem proposta, todos os valores obtiveram um erro menor, que resulta em valores de acurácia maiores. Porém, no treino, os valores obtidos no erro mínimo e no erro médio foram maiores, o que pode significar que o modelo proposto obteve uma maior generalização para a base usada.

Tabela 1 - Dados da Acurácia da Abordagem Proposta e de Cococcioni et al. (2007)

Métricas	Cococcioni et al. (2007)		Abordagem Proposta	
	Sem Pré-Seleção		Com Pré-Seleção	
	Treino	Teste	Treino	Teste
Erro Mínimo	0.3778	0.6162	0.3973	0.5629
Erro Máximo	34.5146	41.1451	25.9483	29.9908
Erro Médio	0.5632	1.9204	0.5751	1.6732
Desvio Padrão	0.5746	1.4250	0.4887	1.2098

Na Tabela 2 (referente à Interpretabilidade) não há a fase de Testes apenas a fase de Treino. Isso ocorre pois, uma vez que o modelo foi obtido no treinamento, a interpretabilidade não se modifica devido ao fato da base de regras já está definida.

Nessa tabela, observa-se que o resultado obtido para o Índice de Interpretabilidade Mínimo foi 0.0055 maior na abordagem de Cococcioni et al. (2007) em relação a abordagem proposta.

Na mesma tabela observa-se que o resultado obtido para o Índice de Interpretabilidade Máximo foi 0.0012 maior na abordagem de Cococcioni et al. (2007) em relação a abordagem proposta.

Ainda na Tabela 2, o resultado obtido para o Índice de Interpretabilidade Médio foi 0,0003 maior em relação a abordagem proposta.

Esses resultados demonstram que não houve mudanças significativas da interpretabilidade obtidas em relação ao modelo com e sem pré-seleção. Nesse caso, apesar da abordagem proposta incluir mecanismos de busca por uma fronteira mais dispersa não houve a diminuição significativa da interpretabilidade da base analisada.

Tabela 2 - Dados da Interpretabilidade da Abordagem Proposta e de Cococcioni et al. (2007)

Métricas	Cococcioni et al. (2007)	Abordagem Proposta
	Sem Pré-Seleção	Com Pré-Seleção
	Treino	Treino
Índice de Interpretabilidade Mínimo	0.3849	0.3794
Índice de Interpretabilidade Máximo	0.5640	0.5628
Índice de Interpretabilidade Médio	0.4804	0.4801

Na Tabela 3 (referete ao Índice de Dispersão) observa-se que o resultado obtido para o Índice Médio de Dispersão foi 0.0029 menor na abordagem de Cococcioni et al. (2007) em relação a abordagem proposta.

Na mesma tabela observa-se, no entanto, que o resultado obtido para a Média do Número de Solucoes foi 0.79 menor na abordagem de Cococcioni et al. (2007) em relação a abordagem proposta.

Esses resultados demonstram que as diferenças no índice de dispersão foram mínimas o que pode significar que para a dispersão das soluções e em relação a base usada não houve mudanças expressivas.

Tabela 3 - Dados do Índice de Dispersão da Abordagem Proposta e de Cococcioni et al. (2007)

Métricas	Cococcioni et al. (2007)	Abordagem Proposta
	Sem Pré-Seleção	Com Pré-Seleção
Índice Médio de Dispersão	1.6245	1.6274
Desvio Padrão do Índice de Dispersão	0.1638	0.1483
Média do Número de Soluções	91.84	92.63

6. Conclusões

O trabalho aqui apresentado trata do desenvolvimento de um Operador de Pré-seleção que atue com foco na diversidade das soluções. A ideia foi a de proporcionar uma maior variabilidade genética o que pode levar à fuga de ótimos locais e, por conseguinte, resulta no aumento da probabilidade de obtenção de melhores soluções.

Conclui-se que, a partir da adição do operador de pré-seleção ao algoritmo (2+2)M-PAES e dos testes comparativos realizados na base *Box-Jenkins Gas Furnace* (BJGF) tem-se que a pré-seleção propicia uma maior diversidade das soluções e uma maior exploração do espaço de busca. Esses fatores corroboram com a fuga de ótimos locais e, por consequência, aumentam a probabilidade de se encontrar melhores soluções.

Como trabalhos futuros pretende-se realizar testes em bases diferentes da BJGF em relação ao número de instancias (a BJGF tem 295 instancias) e número variáveis (BJGF tem 11) e nelas usar conjuntos *Fuzzy* com granularidade diferentes a fim de adaptar o SFBR ao tipo de base a ser testada.

Referências

- Abadeh, Mohammad Saniee, Hamid Mohamadi, and Jafar Habibi. 2011. "Design and Analysis of Genetic Fuzzy Systems for Intrusion Detection in Computer Networks." *Expert Systems with Applications* 38 (6): 7067–75. doi:10.1016/j.eswa.2010.12.006.
- Caridá, V. F., O. Morandin, and C. C. M. Tuma. 2015. "Approaches of Fuzzy Systems Applied to an AGV Dispatching System in a FMS." *The International Journal of*

- Advanced Manufacturing Technology* 79 (1–4). Springer London: 615–25. doi:10.1007/s00170-015-6833-8.
- Cococcioni, Marco, Pietro Ducange, Beatrice Lazzerini, and Francesco Marcelloni. 2007. “A Pareto-Based Multi-Objective Evolutionary Approach to the Identification of Mamdani Fuzzy Systems.” *Soft Computing* 11 (11): 1013–31. doi:10.1007/s00500-007-0150-6.
- Cordón, O., F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena. 2004. “Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends.” *Fuzzy Sets and Systems* 141 (1): 5–31. doi:10.1016/S0165-0114(03)00111-8.
- Deb, K, S Pratab, S Agarwal, and T Meyarivan. 2002. “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II.” *IEEE Transactions on Evolutionary Computing* 6 (2): 182–97. doi:10.1109/4235.996017.
- Fazzolari, Michela, Rafael Alcalá, Yusuke Nojima, Hisao Ishibuchi, and Francisco Herrera. 2013. “A Review of the Application of Multiobjective Evolutionary Fuzzy Systems: Current Status and Further Directions.” *IEEE Transactions on Fuzzy Systems* 21 (1): 45–65. doi:10.1109/TFUZZ.2012.2201338.
- Gacto, M.J., R. Alcalá, and F. Herrera. 2011. “Interpretability of Linguistic Fuzzy Rule-Based Systems: An Overview of Interpretability Measures.” *Information Sciences* 181 (20): 4340–60. doi:10.1016/j.ins.2011.02.021.
- Kim, Eun-Hu, Sung-Kwun Oh, and Witold Pedrycz. 2017. “Reinforced Rule-Based Fuzzy Models: Design and Analysis.” *Knowledge-Based Systems* 119: 44–58. doi:https://doi.org/10.1016/j.knosys.2016.12.003.
- Knowles, J., and D. Corne. 1999. “The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation.” In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 98–105. IEEE. doi:10.1109/CEC.1999.781913.
- . 2000. “Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy.” *Evolutionary Computation* 8 (2): 149–72. doi:10.1162/106365600568167.
- Pimenta, Adinovam H. M., and Heloisa de Arruda Camargo. 2015. “NSGA-DO: Non-Dominated Sorting Genetic Algorithm Distance Oriented.” In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1–8. Istanbul - Turkey: IEEE. doi:10.1109/FUZZ-IEEE.2015.7338080.
- Riid, Andri, and Jürgo-Sören Preden. 2017. “Design of Fuzzy Rule-Based Classifiers through Granulation and Consolidation.” *Journal of Artificial Intelligence and Soft Computing Research* 7 (2). De Gruyter Open: 137–47. doi:10.1515/jaiscr-2017-0010.
- Riza, Lala Septem, Mila Pradini, Eka Fitrajaya Rahman, and Rasim. 2017. “An Expert System for Diagnosis of Sleep Disorder Using Fuzzy Rule-Based Classification Systems.” *IOP Conference Series: Materials Science and Engineering* 185 (1). IOP Publishing: 12011. doi:10.1088/1757-899X/185/1/012011.

Metodologia automática para diagnóstico do glaucoma usando abordagem *Deep Learning*

Marcos Vinícius dos Santos Ferreira¹, Alcilene D. De Sousa¹, Antônio Oseas de C. Filho¹, Patrícia Medyna L. de L. Drumond¹, Patrícia Vieira da S. Barros¹

¹Campus Senador Helvídio Nunes de Barros - Universidade Federal do Piauí (UFPI)
Picos- Piauí - Brasil

{ferreiraviniciussdi,alcileneeluzsousa,antoniooseas}@gmail.com

petibarrosthe@gmail.com, patymedy@hotmail.com

Abstract. *Glaucoma is an asymptomatic disease caused by optic nerve damage due to elevated intraocular pressure. With early diagnosis, the chances of controlling your progression are greater. Glaucoma is a major health problem worldwide as a cause of blindness, only losing to the cataract. This paper presents the development of a method for the automatic detection of glaucoma in retinal images using a deep learning approach. The methodology used in the work was image acquisition, convolutional neural network training for detection of the optic disc, extraction of the blood vessels and extraction of characteristics based on the texture. Finally, classification was performed using convolutional neural networks reaching an accuracy of 98.5%.*

Resumo. *O Glaucoma é uma doença assintomática causada por lesão do nervo óptico em decorrência do aumento elevado da pressão intra-ocular. Com o diagnóstico precoce, as chances de controlar sua progressão são maiores. O glaucoma é um grande problema na saúde mundial como causa de cegueira, perdendo apenas para a catarata. Este trabalho apresenta o desenvolvimento de um método para a detecção automática do glaucoma em imagens da retina utilizando uma abordagem de aprendizado profundo. A metodologia utilizada no trabalho foi aquisição de imagens, treinamento da rede neural convolucional para detecção do disco óptico, extração dos vasos sanguíneos e extração de características baseadas na textura. Por fim, foi realizada a classificação utilizando redes neurais convolucionais alcançando uma acurácia de 98.5%.*

1. Introdução

O glaucoma é uma doença crônica assintomática que devido ao aumento da Pressão Intra-Ocular (PIO) elevada, causa dano ao nervo óptico levando a perda periférica irreversível ou total da visão [Lamani et al. 2014, Yadav et al. 2014]. O nervo óptico é a parte do olho que transmite a informação visual da retina até o cérebro. Com o nervo óptico lesionado, causa um maior alargamento da escavação ou Copo Óptico-(CO), se comparado ao alargamento do Disco Óptico-(DO).

O glaucoma não possui cura e quanto mais cedo o paciente for diagnosticado, menor são as chances de sofrer com os efeitos da doença, uma vez que, o tratamento precoce retardará a evolução da mesma. [Agarwal et al. 2015]. O glaucoma está entre

as principais causas de cegueira do mundo, atingindo aproximadamente 66,8 milhões de pessoas, e estimando que 10% desta população torna-se cega de ambos os olhos devido a esta doença, sendo um grande problema na saúde mundial ficando atrás apenas para catarata [Yin et al. 2012].

Para o diagnóstico, pode ser feito tanto o exame clínico por um oftalmologista como a tonometria, que é um exame oftalmológico que permite a medida PIO, ou usando equipamentos caros especializados como a *Optical Coherence Tomography* (OCT), e *Heidelberg Retina Tomography* (HRT) [Burgoyne et al. 2005]. O OCT é um exame não invasivo que mede a espessura da camada de fibra nervosa da retina, feito através uma tecnologia especial que cria imagens seccionais dos tecidos. O HRT é um scanner a laser confocal que realiza a aquisição e análise de imagens tridimensionais do nervo óptico para detecção de lesões glaucomatosas. No entanto, a avaliação do DO por um oftalmologista é subjetiva e a disponibilidade dos exames OCT e HRT precisam de mão de obra especializada, e possuem um custo elevado [Liu et al. 2011].

Uma alternativa viável a esses exames são os *Computer-Aided Detection* (CAD) / *Computer-Aided Diagnostic* (CADx). Esses sistemas são ferramentas computacionais para auxiliar e melhorar os diagnósticos médicos através de imagens. Assim os CAD e CADx tem por objetivo aumentar a precisão do diagnóstico, proporcionando ao especialista uma segunda opinião, uma vez que informações adicionais resultam em um diagnóstico mais preciso. Assim, neste trabalho foi desenvolvido um sistema CAD/CADx para diagnóstico automático do glaucoma contribuindo para um diagnóstico mais acessível [Ronneberger et al. 2015, de Oliveira et al. 2015]. Para isso, é utilizado uma arquitetura de *Deep Learning*, a *Convolutional Neural Network* (CNN) para detectar as Regiões de Interesse (ROIs) sobre imagens da retina. Em etapas posteriores, são extraídas as características das ROIs utilizando descritores baseados em textura. Por fim são classificadas as informações obtidas com os descritores para quantificação do glaucoma e avaliar o método proposto.

Este trabalho traz contribuições diretas para alguns campos. No campo da medicina, a contribuição é o desenvolvimento de um sistema automático para diagnóstico do glaucoma através de CNN e análise da textura do DO. No campo da ciência da computação, contribuimos nos seguintes aspectos: a) no uso de CNN para reconhecimento preciso do DO, e b) no uso de árvores filogenéticas para caracterização de lesões no nervo óptico.

2. Trabalhos Relacionados

Nesta seção, serão abordados trabalhos que trazem em sua metodologia um mecanismo de auxílio no diagnóstico do glaucoma utilizando técnicas de Processamento Digital de Imagens (PDI), *Deep Learning*, e Reconhecimento de Padrões.

Em Chen *et al.* (2015) os autores desenvolvem uma arquitetura de CNN para ser capaz de capturar as características discriminativas que melhor caracterizam os padrões ocultos relacionados ao glaucoma. A estrutura *Deep Learning* adotada contém seis camadas: quatro camadas convolucionais e duas camadas totalmente conectadas. A saída da última camada totalmente conectada é alimentada para um classificador *soft-max* para a previsão do glaucoma. Os resultados mostram a área sob curva ROC de 0.831 e 0.887.

No trabalho de Sevastopolsky (2017) é apresentado uma abordagem para a

segmentação automática do DO e CO que se baseia em *Deep Learning* utilizando uma modificação da arquitetura de CNN U-Net. O método atinge qualidade comparável aos atuais métodos de última geração para a segmentação do DO e CO, superando-os em termos de tempo de previsão. Os resultados mostram que a segmentação automática do DO pode ser feita com qualidade comparativa com o humano, entretanto o CO é mais complexo para reconhecer.

Em Orlando *et al.* (2017) os autores realizam a análise da viabilidade do uso de CNN que são pré-treinados a partir de dados não médicos para a detecção automatizada de glaucoma. Duas CNNs diferentes pré-treinadas, foram aplicadas em imagens de fundo para afinar os pesos dessa rede, para ter um classificador explicitamente dedicado à nova tarefa de detectar o DO e CO. Os resultados no conjunto de dados Drishti-GS1, avaliados em termos de área sob a curva ROC média, sugerem a viabilidade desta abordagem e oferecem evidências significativas da importância do pré-processamento de imagem bem escolhido para transferência de aprendizagem quando a quantidade de dados não é suficiente para ajustar a rede.

Maninis *et al.* (2016) propõem uma arquitetura de CNN básica DRIU, para segmentação de vasos retinianos e DOs que é rápido e preciso. A DRIU usa uma arquitetura de rede básica na qual dois conjuntos de camadas especializadas são treinados para resolver a separação do disco da retina e do disco óptico. Nos conjuntos de dados testados, DRIU apresenta resultados que não só têm um menor erro mediano, mas também mostram menos dispersão, portanto, mais consistência. Os resultados qualitativos corroboram que a DRIU é robusta e consistente, mesmo nos cenários mais desafiadores e diversos da base de dados RIM-ONE.

Yadav *et al.* (2014) apresentam em seu método uma tentativa de estudar e analisar os recursos de textura em imagens no fundo de olho e suas variações quando a imagem apresenta incidência do glaucoma. As características de textura são localizadas em torno do DO e extraídas com a matriz de co-ocorrência. O método de classificação proposto é o uso do classificador de rede neural com a ajuda da extração de características de textura da área localizada no DO das imagens do fundo de olho. O método de classificação de 72% de acurácia em classificar imagens normais ou glaucomatosas.

Diante do exposto nos trabalhos relacionados, o problema é trazer um diagnóstico mais completo, e torná-lo mais eficiente para que possa ser utilizado como sistema de diagnóstico auxiliado por computador, que ajudam a diminuir a variabilidade de interpretação, fornecendo uma segunda opinião ao especialista. Este trabalho tem como objetivo a utilização da *Deep Learning* para identificar ROIs de imagens da retina para serem posteriormente diagnosticadas quanto a presença do glaucoma.

3. Materiais e Métodos

Esta seção aborda a metodologia proposta neste trabalho para detecção, extração, e classificação de características do DO, a fim de diagnosticar automaticamente o glaucoma a partir de imagens digitais da retina.

A primeira etapa foi a aquisição de imagens, utilizando três bases de imagens públicas de fundo de olho. Em seguida, foram reunidas todas as imagens da retina, com olhos normais e em diferentes níveis de glaucoma para o treinamento da rede neural de

aprendizado profundo. A rede utilizada é uma implementação da Rede Neural Convolutiva (CNN) baseada na arquitetura U-Net [Ronneberger et al. 2015], empregada com o objetivo de realizar o reconhecimento de padrões para detectar o DO.

Concluído o treinamento, é realizado com os pesos treinados da CNN a predição sobre as imagens de teste para segmentação da imagem, a fim de identificar a máscara com a ROI. Com a máscara da ROI definida, foi realizada a sobreposição da imagem testada com a máscara que a CNN detectou, para ter somente a região do DO a fim de facilitar a análise sobre as características da patologia e diminuir a região a ser processada. Em seguida foi realizada a extração dos vasos sanguíneos através de uma segmentação por limiar, pois os vasos não são relevantes para as próximas etapas de análise.

Após essa etapa, foi feita a extração de características na tonalidade de cinza das imagens, utilizando descritores de textura baseados em índices de diversidade filogenética. Por fim, foi realizada a classificação com um classificador baseado em CNN, para verificar a eficácia do método ao quantificar as imagens em normais ou glaucomatosas. Na Figura 1 tem-se o fluxograma que representa as etapas propostas na metodologia para diagnóstico do glaucoma.

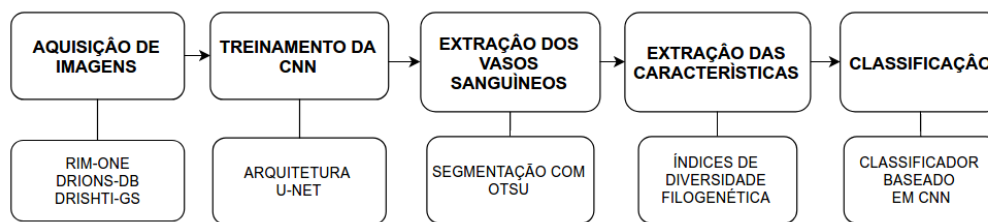


Figura 1. Fluxograma da metodologia proposta.

3.1 Aquisição de Imagens

Neste trabalho são utilizadas três bases de imagens da retina, a RIM-ONE [Trucco et al. 2013], a DRIONS-DB [Carmona et al. 2008] e DRISHTI-GS [Sivaswamy et al. 2014]. Essas são bases de imagens públicas que foram dispostas para pesquisa, teste e desenvolvimento de soluções computacionais para auxílio ao diagnóstico do glaucoma. A base RIM-ONE possui 169 imagens sendo 119 de retinas normais, e as outras 51 com níveis diferentes do glaucoma, possuindo todas cinco marcações do DO para cada imagem. A base DRISHTI-GS contém 101 imagens, sendo que 50 delas apenas possuem a marcação do DO feito por especialistas. Já a base DRIONS-DB contém 110 imagens, as marcações do DO foram feitas por dois especialistas.

3.2 Convolutional Neural Network

Convolutional Neural Network é uma arquitetura de *Deep Learning*, sendo uma variação da rede neural artificial perceptron de múltiplas camadas, baseada no processamento de dados visuais, capaz de aplicar filtros nesses dados, mantendo a relação de vizinhança entre os *pixels* da imagem ao longo do processamento da rede [LeCun et al. 1998]. O nome

“Convolutional Neural Network” indica que a rede emprega uma operação matemática chamada convolução.

Em técnicas de *Deep Learning*, existem várias arquiteturas de CNNs, cada uma desenvolvida com um objetivo, seja classificação, como a GoogLeNet [Szegedy et al. 2015], detecção como VGG Net [Simonyan and Zisserman 2014], reconhecimento como AlexNet [Krizhevsky et al. 2012], entre outras. Para tratar a primeira etapa do problema proposto, buscou-se utilizar uma arquitetura de CNN para segmentação de imagens. Para detecção do ROI, a imagem da retina tem que ser segmentada para separar o DO do restante da imagem. Para isso utilizou-se a arquitetura de CNN U-net, que tem atingido bons resultados em segmentação rápida e precisa de imagens médicas. Ela vem superando os métodos existentes em vários desafios biomédicos de segmentação de imagens [Ronneberger et al. 2015]. A Figura 2 ilustra a arquitetura e as camadas que constituem a rede.

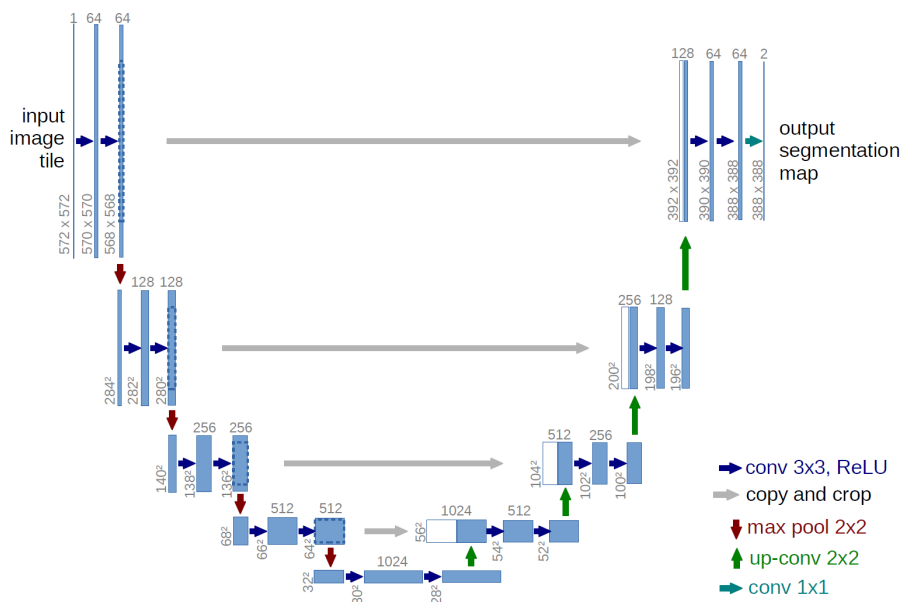


Figura 2. Representação da arquitetura U-net.

Como ilustrado na Figura 2, cada caixa azul corresponde a um mapa de recursos multicanal. A quantidade de canais é indicada no topo da caixa. O tamanho x-y é fornecido na margem inferior esquerda da caixa. As caixas brancas representam mapas de recursos copiados. As setas indicam as diferentes operações. No total, a rede possui 23 camadas convolutivas. Como algoritmo de otimização da rede, utilizou-se algoritmo *Adam*, pois é invariável ao redimensionamento diagonal dos gradientes e é adequado a grandes problemas em termos de dados.

A arquitetura consiste em dois caminhos, um de contração (lado esquerdo) e o outro um caminho expansivo (lado direito). O caminho de contração segue a arquitetura típica de uma CNN. Consiste na aplicação repetida de duas convoluções 3x3, cada uma seguida por uma unidade linear retificada (ReLU) e uma operação de *max-pooling* de 2x2 com *stride* 2 para *downsampling*. Em cada etapa de *downsampling* dobra-se o número de canais de recursos.

Cada passo no caminho expansivo consiste em um levantamento do mapa de recursos seguido de uma convolução 2x2 que faz a metade do número de canais de recursos, que é uma junção com o mapa de recursos correspondentemente cortado do caminho de contratação e dois 3x3 circunvoluções, cada uma seguida por uma função de ativação uma ReLU. Essa etapa do caminho expansivo é necessário devido à perda de pixels de borda em cada convolução no caminho de contratação. Na camada final é utilizada uma convolução 1x1 para mapear cada vetor de características de 64 componentes para o número desejado de classes.

3.3 Treinamento

Para o treinamento, os algoritmos foram desenvolvidos em linguagem de programação Python 3.5, e *framework* Keras [Keras 2017] para treinamento de redes neurais (com TensorFlow *backend* [Abadi et al. 2016]). A biblioteca OpenCV [Bradski and Kaehler 2008] foi usada para manipulação de imagens. A máquina utilizada contém hardware composto por processador Intel Core I5 64 bits, e 4GB de memória RAM DDR3.

Com a CNN implementada, o próximo passo e que merece muita atenção, é selecionar os dados de entrada para o treinamento da rede, pois são com eles que a CNN aprende. Para isso, foram selecionadas aleatoriamente 326 imagens das três bases, sendo 155 da RIM-ONE, 51 da DRISHTI-GS, e 120 da DRIONS-DB, e separou-se 65 imagens para testes, sendo estas, também compostas por imagens das três bases e escolhidas aleatoriamente. Foram utilizadas imagens da retina e suas máscaras para o treinamento, pois as máscaras apontam as saídas desejadas sobre as imagens da retina, ou seja, a rede irá aprender a localizar o DO em imagens da retina.

[Dosovitskiy et al. 2014, Maninis et al. 2016] relatam que redes neurais convolucionais são eficientes no aprendizado com uma grande quantidade de dados de entrada. Em posse dessa afirmação e observando o pequeno número de imagens disponíveis para treinamento, foi adotado como alternativa a rotação de cada uma das 326 imagens nos ângulos: 22, 45, 67, 90, 112, 135, 157, 202, 225, 247, 270, 290, 315, e 337 graus juntamente com suas máscaras, produzindo para cada imagem 14 novas imagens rotacionadas, resultando em 4564 imagens para treinamento. O treinamento realizado é a partir do zero, ou seja, nenhum modelo pré-treinado foi usado e os pesos dos filtros convolucionais foram iniciados aleatoriamente no início de cada treino.

Para a rede CNN aprender as características e padrões do DO, que é a região de interesse, foram passados como parâmetros de entrada as imagens da retina e suas devidas máscaras, sendo o papel da máscara apontar a classe que deve ser aprendida, que no caso é toda a estrutura do DO.

3.4 Extração dos Vasos Sanguíneos

Antes de extrair as características das imagens segmentadas com a CNN, retirou-se os vasos sanguíneos para facilitar a abstração de características relevantes para posterior classificação. Para isso, utilizou-se o algoritmo de segmentação por limiar de Otsu. Este algoritmo irá permitir-nos obter de forma rápida e automática o valor limite correto para escolher entre dois modos de histograma, permitindo-lhes aplicar o limiar de forma otimizada. Este algoritmo tenta encontrar um valor limiar (t) que minimiza a variância ponderada dentro da classe dada pela relação. Otsu binariza a imagem, como os vasos

sanguíneos possuem uma tonalidade mais escura no DO, prevalece somente a região do DO sem os vasos.

3.5 Extração de Características

A diversidade filogenética procura as diferentes espécies ou de indivíduos diferentes em uma população, provavelmente começará com uma análise filogenética. Para a extração de características das ROIs após a detecção do DO e extração das veias, utilizou-se os índices de diversidade filogenéticas *Phylogenetic Diversity* (PD), *Sum of Phylogenetic Distances* (SPD), *Mean Nearest Neighbour Distance* (MNND), *Phylogenetic Species Variability* (PSV) e *Phylogenetic Species Richness* (PSR) para extração de características baseadas em textura. Esses índices consideram a distância filogenética, calculada a partir da arquitetura da árvore filogenética.

Para isso, é feito uma analogia das propriedades da filogenia da biologia com as propriedades de uma imagem. O que representa uma comunidade na biologia, na metodologia vai representar uma ROI da imagem da retina. As espécies, são comparadas a quantidade de níveis de cinza. Os indivíduos, são representados como os pixels da imagem. E a distância filogenética é o número de arestas entre duas espécies. Assim são calculados os índices sobre as imagens para procurar e descrever essa relação filogenética, para posterior classificação.

O índice PD é o somatório dos comprimentos dos ramos da filogenia de cada espécie. Na Equação 1, o B é o número de ramificação da árvore, L_i é o comprimento do ramo e A_i é a abundância média de espécies que compartilham ramo i .

$$B \times \frac{\sum_i^B L_i A_i}{\sum_i^B A_i} \quad (1)$$

O SPD é a soma das distâncias filogenéticas entre cada par de espécies. O cálculo do SPD pode ser observado na Equação 2, onde d_{mn} é a distância entre as espécies m e n ; a_m é a abundância de espécies m ; S o número de espécies no conjunto focal.

$$\left(\frac{S(S-1)}{2} \right) \times \frac{\sum \sum m < n^2 m n^a m^a n}{\sum \sum m < n^a m^a n} \quad (2)$$

O MNND é a distância média do táxon mais próximo. Na Equação 3, onde d_{mn} é a distância entre as espécies m e n ; a_m , abundância de espécies m .

$$\sum_m^S m i n (d_{mn}) a_m \quad (3)$$

O PSV é a variabilidade de espécies filogenéticas, que resume o grau em que as espécies em uma comunidade são filogeneticamente relacionadas. A variabilidade de espécies filogenéticas, resume o grau em que as espécies em uma comunidade são filogeneticamente relacionadas. O cálculo do PSV é definido na Equação 4, onde, o trC representa a soma dos valores da diagonal de uma matriz C , $\sum c$ é o somatório de todos os valores da matriz, n é o número de espécies e \bar{c} é a média dos elementos da diagonal de C .

$$PSV = \frac{ntrC - \Sigma c}{n(n-1)} = 1 - \bar{c} \quad (4)$$

E o PSR é a riqueza de espécies e quantifica o número de espécies em uma comunidade. A riqueza de espécies filogenéticas (PSR), quantifica o número de espécies em uma comunidade. O valor da riqueza de espécies filogenéticas é encontrado multiplicando-se o número de espécies n pela variabilidade da comunidade, pode-se verificar os parâmetros na Equação 5.

$$PSR = nPSV \quad (5)$$

3.6 Classificação

Para a classificação foi utilizada a suíte de algoritmos de mineração de dados, pré-processamento de dados, classificação, visualização, e aprendizado de máquina WEKA (*Waikato Environment for Knowledge Analysis*) [WEKA 2017]. Utilizou-se como classificador uma implementação baseada na última camada das CNNs, a *fully-connected*, sendo essa própria para trabalhar com números gerados pelos índices de diversidade filogenéticas [Amten 2017]. Esse classificador foi utilizado por além de classificar os resultados, aprende com os dados de treino, e utiliza esse aprendizado nos dados de teste, obtendo um bom resultado na categorização de padrões do DO.

3.7 Validação dos Resultados

Para validação dos resultados do método em quantificar o glaucoma em imagens de fundo de olho, utilizou-se de métricas de avaliação baseadas em estatística como Acurácia, Especificidade, Sensibilidade, Precisão, e Sobreposição [Metz 1986, Martinez et al. 2003].

4. Resultados e Discussões

Nesta seção são apresentados os resultados obtidos na metodologia proposta em classificar as imagens em glaucomatosas ou normais, com base nas imagens de teste. Em seguida, foi calculada a acurácia, sensibilidade, precisão, sobreposição e especificidade para cada teste, e por fim é feito a análise comparativa com outros trabalhos relacionados.

4.1 Configuração dos Testes

Para avaliar os resultados, foi realizado um teste selecionando aleatoriamente 14 imagens da base RIM-ONE e 51 da base DRISHTI-GS. Dentre elas, contém imagens da retina normais e com diferentes níveis de glaucoma, sendo que essas imagens não foram utilizadas para treinamento, somente para teste. Com isso desejava-se saber se após o treino, a CNN conseguiria identificar o DO em novas imagens da retina para garantir que a CNN aprendeu a reconhecer padrões do DO. Essas duas bases foram utilizadas para teste por terem a descrição identificando quem são imagens normais e glaucomatosas segundo os especialistas.

A base DRIONS-DB apresenta 23,1% das imagens glaucomatosas, mas mediante pesquisas não foram encontradas as identificações de quais imagens da base são glaucomatosas, por isso a base não foi utilizada para teste. A mesma só foi utilizada para treinamento por possuir a máscara do DO.

4.2 Resultados dos Testes

Na Figura 3, apresentam os resultados sobre as etapas para processamento da ROI. Na Figura 3(a) tem-se a imagem utilizada para teste. Na Figura 3(b) tem-se a predição feita pela CNN, realizando a segmentação sobre essa imagem, retornando a máscara apontando o DO. Na Figura 3(c) é ilustrado a sobreposição feita para ter a ROI isolada. Por fim, na Figura 3(d) ilustra a imagem resultante após a extração dos vasos sanguíneos.

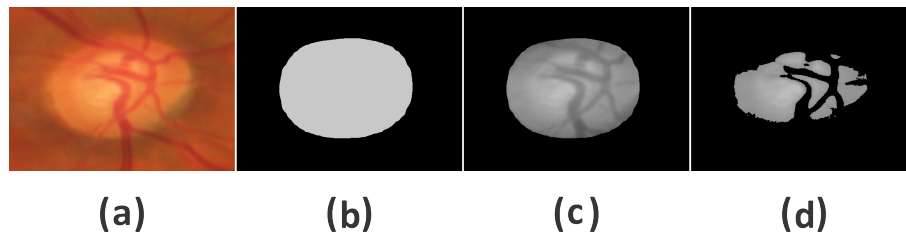


Figura 3. Detecção do DO.

Com os vasos sanguíneos extraídos, foram extraídas as características com descritores de textura. Para isso foram utilizados os índices de diversidade filogenética PD, SPD, MNND, PSV, PSR, com a finalidade de descrever informações a partir dos níveis de cinza da ROI. Em seguida foi realizada a classificação executando o WEKA com um classificador baseado em CNN, sobre as características extraídas das imagens. Tal classificador reconhece padrões sobre as informação extraídas pelos índices, distinguindo regiões e classificando imagens em normais ou doentes, trazendo um diagnóstico.

De acordo com as informações extraídas pelos índices, as ROIs das imagens normais apresentam características mais homogêneas em relação as ROIs das imagens glaucomatosas, que apresentam características heterogêneas devido as lesões, sendo a variabilidade de espécies, níveis de cinza, maior nas ROIs em imagens glaucomatosas. Os resultados obtidos na classificação foram promissores, o classificador da rede neural CNN atingiu uma acurácia de 98,5%, sensibilidade de 98,1%, especificidade de 100%, precisão de 100% e sobreposição de 93% em diagnosticar imagens normais e glaucomatosas.

Ao utilizar novas imagens da retina no método proposto, os resultados podem alcançar uma maior precisão na detecção do DO em diferentes amostras. Isso se deve ao fato que o núcleo do trabalho se baseia no aprendizado de máquina profundo, quanto mais amostras, maior a curva de aprendizado e precisão para reconhecimento dessas estruturas.

Tabela 1. Comparação com os trabalhos relacionados

Trabalho	Base	A (%)
[Sevastopolsky 2017]	RIN-ONE, DRIONS-DB, DRISHTI-GS	-
[Chen et al. 2015]	ORIGA, SCES	-
[Orlando et al. 2017]	DRISHTI-GS1, DRIVE	-
[Maninis et al. 2016]	DRIVE, STARE	-
[Yadav et al. 2014]	Privada	72
Metodologia	RIN-ONE, DRIONS-DB, DRISHTI-GS	98.5

Em comparação com os trabalhos relacionados, como demonstra a Tabela 1, a metodologia se destacou. Os trabalhos de [Sevastopolsky 2017, Maninis et al. 2016]

propuseram em suas metodologias o uso de técnicas de aprendizado profundo, com o uso de CNNs para detecção do DO, OC e vasos sanguíneos, mas as metodologias apenas trabalham na parte de segmentação das imagens de retina. Nos trabalhos de [Chen et al. 2015, Orlando et al. 2017] os autores em suas metodologias vão além da segmentação, eles quantificam a patologia, mas a abordagem e valores dos resultados se diferem aos usados na metodologia proposta por a mesma utilizar os descritores baseados em textura para extrair características da imagens. No trabalho de [Yadav et al. 2014] os autores obtém 72% de acurácia na classificação do glaucoma, mas utilizam a matriz de concorrência para extrair características.

O referente trabalho se difere dos demais por utilizar uma metodologia mais completa e complexa para o diagnóstico do glaucoma, apresentando uma abordagem completa, automatizando o diagnóstico da patologia. A detecção do DO e OC para obter a relação do diâmetro da escavação e do disco (*cup-to-disk diameter ratio-CDR*), é indício para quantificar o glaucoma conforme relatado e empregado nos trabalhos relacionados. A metodologia proposta usou a CNN para detectar o DO e classificar as características, em que foram utilizados os descritores de textura para extrair as características do DO, com intuito de obter informações que quantifiquem a patologia com classificadores, sem necessidade então de detectar o CO e calcular o CDR.

5. Conclusão

Deep Learning é uma subárea da Inteligência Artificial que está em crescente avanço, inovando com o surgimento em aplicações de diversas técnicas e usos, impulsionando uma nova era de automatização inteligente. Com o uso de técnicas de PDI e IA, esse trabalho apresentou uma metodologia para detecção automática do DO em imagens da retina com CNN, utilizando descritores baseados em textura e o classificador CNN para discriminação e classificação de regiões em imagens da retina em glaucomatosas ou normais.

A análise de imagens da retina utilizando a ROI apresentou resultados eficientes na categorização de padrões de lesões glaucomatosas, atingindo uma taxa de acerto de 98.5% de acurácia, sensibilidade de 98,1%, especificidade de 100%, precisão de 100%, e sobreposição de 93% na quantificação do glaucoma. Assim, o referente trabalho tem uma maior relevância na área médica e computacional, pois com o uso dessas novas tecnologias, possibilitam ao profissional da área recursos para diagnosticar a doença fornecendo um diagnóstico mais precoce.

A partir dos resultados obtidos, pode-se inferir que a utilização da CNN para detecção da ROI, e descritores baseados na textura das regiões, apresentam resultados eficazes na classificação do quantificação do glaucoma. O uso de índices filogenéticos para descrever padrões em regiões de imagens mostrou-se eficiente para a metodologia proposta.

Referências

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

- Agarwal, A., Gulia, S., Chaudhary, S., Dutta, M. K., Burget, R., and Riha, K. (2015). Automatic glaucoma detection using adaptive threshold based technique in fundus image. In *Telecommunications and Signal Processing (TSP), 2015 38th International Conference on*, pages 416–420. IEEE.
- Anten, J. (2017). <https://github.com/anten/neuralnetwork>.
- Bradski, G. and Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. "O'Reilly Media, Inc."
- Burgoyne, C. F., Downs, J. C., Bellezza, A. J., Suh, J.-K. F., and Hart, R. T. (2005). The optic nerve head as a biomechanical structure: a new paradigm for understanding the role of iop-related stress and strain in the pathophysiology of glaucomatous optic nerve head damage. *Progress in retinal and eye research*, 24(1):39–73.
- Carmona, E. J., Rincón, M., García-Feijóo, J., and Martínez-de-la Casa, J. M. (2008). Identification of the optic nerve head with genetic algorithms. *Artificial Intelligence in Medicine*, 43(3):243–259.
- Chen, X., Xu, Y., Wong, D. W. K., Wong, T. Y., and Liu, J. (2015). Glaucoma detection based on deep convolutional neural network. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 715–718. IEEE.
- de Oliveira, F. S. S., de Carvalho Filho, A. O., Silva, A. C., de Paiva, A. C., and Gattass, M. (2015). Classification of breast regions as mass and non-mass based on digital mammograms using taxonomic indexes and svm. *Computers in biology and medicine*, 57:42–53.
- Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 766–774.
- Keras (2017). <https://keras.io/>.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- Lamani, D., Manjunath, T., Mahesh, M., and Nijagunarya, Y. (2014). Early detection of glaucoma through retinal nerve fiber layer analysis using fractal dimension and texture feature. *International Journal of Research in Engineering and Technology*.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Liu, J., Yin, F., Wong, D., Zhang, Z., Tan, N., Cheung, C., Baskaran, M., Aung, T., and Wong, T. (2011). Automatic glaucoma diagnosis from fundus image. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 3383–3386. IEEE.
- Maninis, K.-K., Pont-Tuset, J., Arbeláez, P., and Van Gool, L. (2016). Deep retinal image understanding. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 140–148. Springer.

- Martinez, E. Z., Louzada-Neto, F., and Pereira, d. B. B. (2003). A curva roc para testes diagnósticos. *Cad Saúde Coletiva*, 11(1):7–31.
- Metz, C. E. (1986). Roc methodology in radiologic imaging. *Investigative radiology*, 21(9):720–733.
- Orlando, J. I., Prokofyeva, E., del Fresnob, M., and Blaschko, M. (2017). Convolutional neural network transfer for automated glaucoma identification. In *12th International Symposium on Medical Information Processing and Analysis*, pages 101600U–101600U. International Society for Optics and Photonics.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer.
- Sevastopolsky, A. (2017). Optic disc and cup segmentation methods for glaucoma detection with modification of u-net convolutional neural network. *arXiv preprint arXiv:1704.00979*.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Sivaswamy, J., Krishnadas, S., Joshi, G. D., Jain, M., and Tabish, A. U. S. (2014). Drishti-g: Retinal image dataset for optic nerve head (onh) segmentation. In *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, pages 53–56. IEEE.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9.
- Trucco, E., Ruggeri, A., Karnowski, T., Giancardo, L., Chaum, E., Hubschman, J. P., Al-Diri, B., Cheung, C. Y., Wong, D., Abramoff, M., et al. (2013). Validating retinal fundus image analysis algorithms: Issues and a proposal validating retinal fundus image analysis algorithms. *Investigative ophthalmology & visual science*, 54(5):3546–3559.
- WEKA (2017). Machine learning group at the university of waikato.
- Yadav, D., Sarathi, M. P., and Dutta, M. K. (2014). Classification of glaucoma based on texture features using neural networks. In *Contemporary Computing (IC3), 2014 Seventh International Conference on*, pages 109–112. IEEE.
- Yin, F., Liu, J., Wong, D. W. K., Tan, N. M., Cheung, C., Baskaran, M., Aung, T., and Wong, T. Y. (2012). Automated segmentation of optic disc and optic cup in fundus images for glaucoma diagnosis. In *Computer-based medical systems (CBMS), 2012 25th international symposium on*, pages 1–6. IEEE.

Impact of Base Partition Selection on Multi-Objective Clustering Ensemble

Gabriel Leonardo Pedote¹, Katti Faceli¹, Tiemi Christine Sakata¹

¹Department of Computing
Federal University of São Carlos (UFSCar) — Sorocaba, SP – Brazil

{gpedote,katti,tiemi}@ufscar.br

Abstract. *Unsupervised data clustering is not a trivial process, as no previous knowledge is available and real data is often complex and multi-faceted. To tackle some of these issues, recent efforts that build knowledge considering multiple partitions as base, such as ensemble clustering, emerged. However, special care must be taken in the composition of those partitions, as their quality and diversity proved to be closely related to their performances. To enhance these characteristics, a number of methods to evaluate and select a subset of the partitions have been proposed and successfully applied. In this work, we expand this discussion by evaluating the impact of some of the state-of-the-art selection methods in the novel context of multi-objective cluster ensemble.*

1. Introduction

Traditionally, clustering aims to describe the data being explored under a single perspective. However, it is broadly known that in several cases this approach imposes serious limitations on what could be extracted with the analysis. First, because, in real cases, data is often complex and can be naturally partitioned in many ways [Bailey 2013]. Second, because there is no guarantee that a clustering algorithm will result in a relevant descriptive data model [Sakata et al. 2010]. Furthermore, changes in parameters and preprocessing techniques (e.g., feature selection) can dramatically change the final result, either by evidencing or hiding a possible plural meaning presented in the data [Jain and Dubes 1988, Muller et al. 2010].

Aware of these problems and the recognized subjectivity of the whole process, several works aim the production of multiple solutions, as in multiple clustering [Muller et al. 2010] or consider multiple partitions in order to produce a more stable and accurate single consensus result, as in ensemble clustering [Strehl and Ghosh 2003].

Several works, in both approaches, strongly suggest that the quality and diversity of the multiple considered clusterings exert direct influence in the accuracy and quality of their final result(s) [Zimek and Vreeken 2015, Fern and Lin 2008, Muller et al. 2010, Liu and Fan 2012, Lei et al. 2016]. Diversity is advantageous, in most part, because considering multiple identical — or nearly identical — partitions would likely lead to the same problems faced when considering a unique data interpretation. Quality, on the other hand, is desired because low quality clusterings would have little to add to a better data understanding. It is important to note, though, that quality in unsupervised learning is a loosely defined concept, and it can be defined in many ways [Fern and Lin 2008].

In order to improve these two factors — quality and diversity —, and likely the quality of the final result(s), a selection of partitions that, to certain degrees, comply with these factors within a larger pool of individual clusterings can be applied [Lei et al. 2016, Fern and Lin 2008, Naldi et al. 2013]. Experimentally, the results produced using a selected set of partitions proved to be similar or better than those based on the full set in several cases [Sakata et al. 2010, Lei et al. 2016, Fern and Lin 2008, Naldi et al. 2013].

Multiple selection strategies were evaluated in different clustering contexts so far. [Fern and Lin 2008], assessed, using a number of partition selection methods, how quality and diversity could affect the performance of cluster ensembles. [Sakata et al. 2010], proposed a partition selection method to reduce the number of solutions generated by Pareto-based multi-objective genetic algorithms. [Naldi et al. 2013] expanded the work of [Fern and Lin 2008] evaluating more selection methods with a larger number of ensemble algorithms and datasets. [Lei et al. 2016], also build upon [Fern and Lin 2008] ideas, but took a different approach and evaluated how partition selection could affect meta-clustering. In all of these works, performance gains were obtained with the usage of partition selection.

Based on these findings, in the present work we also investigate the impact of these selection methods. However, we evaluate how some of the selections proposed in [Fern and Lin 2008, Sakata et al. 2010, Naldi et al. 2013] can affect the multi-objective clustering ensemble domain. An analysis not done so far, to the best of our knowledge. We specifically focus on the Multi-Objective Clustering Ensemble (MOCLE) algorithm, that embraces various concepts from multiple clustering and ensemble clustering [Faceli et al. 2009] — areas where those selections were previously analyzed. Our main aim is to present a preliminary case study showing the potential of quality improvement of MOCLE’s clustering results by feeding it with the appropriate selection of base partitions.

The remainder of this paper is organized as follows. Section 2, reviews the main partition selection related approaches. Section 3 summarizes the experimental setup used in this paper, including a description of the selection strategies considered, the dataset employed and MOCLE’s parameterization. The results are presented and discussed in Section 4. Finally, Section 5 summarizes the conclusions and highlights future directions of research.

2. Related Work

In the clustering literature, there are several works where selection strategies were employed to explore the possibility of improvement in their results. Their main purpose is to consider just a subset (selected set) of a partitions’ set (base partitions), with the intent that the usage of those in further processes, instead of the full set, results in equal or better performance. This section describes the selection methods related to ensemble clustering — known as Cluster Ensemble Selection (CES) — and an additional selection strategy from the multi-objective area.

2.1. Selection on Ensemble Clustering

In [Kuncheva and Hadjitodorov 2004], diversity in the base partitions was found to be a direct influence on the accuracy of a cluster ensemble. The work by [Hadjitodorov et al. 2006] corroborated this finding, observing that moderate diversity led to ensembles that

are more accurate. Later, [Fern and Lin 2008], proposed a quality and a diversity measure, and investigated, using a series of partition selection methods, how these two properties could affect cluster ensemble performance. Some of the employed CES resulted in statistically significant gains over full ensembles (i.e., those that considered the full set in the ensemble).

The use of CES was also explored in [Azimi and Fern 2009], where a selection framework was proposed to avoid consensus partitions excessively different from those that they result from. Statistically significant gains over full ensembles were also obtained with this approach.

Another direction to CES investigations was proposed by [Naldi et al. 2013], where a number of relative clustering validation indexes, both individually and combined, were used to assess quality and diversity — these factors until this point were only based on external validity indexes. CES usage was also evaluated with more consensus functions and a greater number of datasets, reinforcing the use of selection for results improvement.

Recently, in [Rastin and Kanawati 2015], another partition selection method was proposed, with the use of a community detection algorithm on a proximity-graph created over the set of base clusterings properties. Improvements were also obtained.

2.2. Independent Selection

A partition selection method with the goal of reducing the number of solutions generated by Pareto-based multi-objective genetic algorithms, was proposed in [Faceli et al. 2010]. This selection aimed to simplify the manual analysis of these outputted solutions by a data specialist, which could contain hundreds of partitions in some cases. Continuing this work, in [Sakata et al. 2010], this method was changed to incorporate two main modifications: (i) the immediate consideration of partitions that most appear in the initial set as part of the resulting selected set; (ii) the automatic adjustment of the internal parameter that indicate when the selection limit was reached. The modified method was named Automatic Selection Algorithm (ASA).

3. Experimental Design

In order to give an insight on the impact of selection in the performance of MOCLE, experiments were carried out using a well-known dataset and following the steps illustrated in Figure 1. First, we generated a pool of partitions using different clustering algorithms to serve as base partitions for our experiments, as described in Section 3.2. From here on, this pool is referred to as the initial pool. Then, each of the selections presented in Section 3.3 was individually applied to the initial pool and produced a reduced set of base partitions for MOCLE's processing¹ — MOCLE is introduced in Section 3.4. After that, MOCLE was run with each reduced set. Due to its non-deterministic nature, for each input MOCLE was run 30 times, producing a set of results for each selection. The conduction of the experiments are described in Section 3.5. The sets of outputs were later compared, assessing the impact of each selection on MOCLE (Section 3.6).

¹MOCLE can be executed online at <http://lasid.sor.ufscar.br/mocleproject/>. All executions in the present work used MOCLE's default configurations.

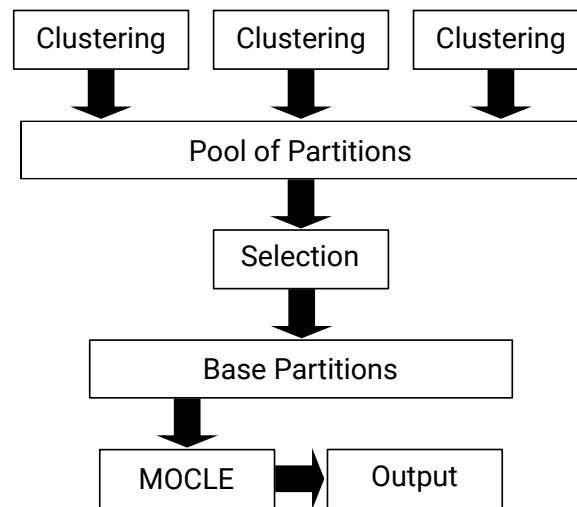


Figure 1. Selection experiments' pipeline.

3.1. The dataset

The `golub` dataset [Golub et al. 1999] used in our experiments, is composed of gene expression data of 72 leukemia cancer patients, obtained at the time of diagnosis. In [Golub et al. 1999], 47 patients were identified with Acute Myeloid Leukemia (AML) and 25 patients with Acute Lymphoblastic Leukemia (ALL), representing one of its known structures. Furthermore, an additional subcategorization of type ALL in two different types of cells (9 T-cell ALL and 38 B-cell ALL) was identified as the other known structure. Therefore, this dataset consists of two known structures, one with two clusters (AML and ALL) and another with three clusters (AML, T-cell ALL AND B-cell ALL), respectively identified as `k2` and `k3` from now on.

This dataset was chosen, because it is one of the most widely cited microarray dataset in the clustering literature. In addition, during our experiments MOCLE had problems in finding the two known structures contained on the data. Our experiments use the same preprocessing phase described in [Dudoit et al. 2002], so `golub`'s number of features was reduced from 7129 to 3571.

3.2. Partition Pool Generation

The partitions were produced considering the number of clusters, k , ranging from 2 to $\sqrt{72} = 8$ (the number was rounded), where $n = 72$ is the number of objects contained in the `golub` dataset. Four clustering algorithms were used: K-Means (KM), Average-Link (AL), Single-Link (SL) [Jain and Dubes 1988] and Shared Nearest Neighbors (SNN) [Ertöz et al. 2002]. For AL and SL, the hierarchy was generated and cut, in order to produce a partition for each value of k . In order to minimize suboptimal solutions, KM was run 30 times for each k and the one with the lowest squared error was chosen. SNN was executed with a number of specific algorithm parameters and produced several partitions. From the partitions created by SNN, we selected 14 partitions with k in the interval of interest. This setting produced the total of 35 partitions. These 35 partitions compose our *initial pool*.

All the implementations used, except SNN², came from the software Cluster 3.0³ [De Hoon et al. 2004].

3.3. Selection Strategies

In this section, we review the selection methods considered in our experiments. First, two groups of CES from [Naldi et al. 2013] work are presented in Section 3.3.1 and Section 3.3.2. Then, the Diversity selection proposed by [Fern and Lin 2008] is discussed in Section 3.3.3. Later, ASA's [Sakata et al. 2010] selection strategy is detailed in Section 3.3.4.

3.3.1. Single Index Selection (SIS)

The main idea of this approach is to select the s -top ranked partitions according to a given clustering relative validation index [Naldi et al. 2013]. Where, s is a parameter corresponding to number of partitions to be selected. Six relative clustering validity measures were adopted as quantitative criteria for evaluating the quality of the partitions. All of them are based on maximizing cluster external isolation and internal cohesion, that is, they prefer compact and separated clusters [Vendramin et al. 2010]. Their difference lies on how this calculation is made and how each measure favors these characteristics:

- Dunn [Vendramin et al. 2010];
- Simplified Silhouette (SS) [Vendramin et al. 2010];
- Alternative Simplified Silhouette (ASS) [Vendramin et al. 2010];
- Calinski-Harabasz (VRC) [Vendramin et al. 2010];
- PBM [Vendramin et al. 2010]; *and*
- Davies-Bouldin (DB) [Vendramin et al. 2010].

3.3.2. Combination of Relative Indexes (CRI)

As pointed in [Naldi et al. 2013] the approach of using a particular index for all the situations, described in Section 3.3.1, has its drawbacks. In first place, it is hard to select an index for a situation in which — in most practical cases — there is no information about their expected performances. Second, the usage of the same criterion for selecting all the partitions, could lead to a biased result, where the best evaluated partitions are likely to have similar characteristics. So, in order to cope with these difficulties a committee of relative validation indexes (CRI) would be a better alternative, expecting that a good performance of most indexes would compensate the weak performance of some [Naldi et al. 2013]. This committee can select the partitions voting in different ways:

- The Sum of Ranks (SR) method works similarly to the SIS selection, but it considers all the six validation indexes; this is done building a ranking for each index based on the evaluation of all base partitions. After that, the sum of the individual rankings is calculated for each base partition (over the different indexes), then the base partitions with the lowest sums are selected.

²The SNN implementation was the same used in [Faceli et al. 2010].

³Available in <http://bonsai.hgc.jp/~mdehoon/software/cluster/software.htm>

- The Best Rank Position (BRP) selection also considers all validation indexes and builds the partition ranking for each index. However, instead of a summation of rankings, this strategy considers each ranking in turns. For example, the first selected partition would be the best ranked according the first index; the second one would be the best according to index two, and so on.
- The Sum of Ranks with Diversity (SRD) method works like SR, but with the additional of also considering the diversity of each base partition, calculated as in [Fern and Lin 2008].

3.3.3. Diversity

The Diversity selection, defined in [Fern and Lin 2008], uses the information shared between the partitions to measure its quality and diversity. Quality was defined as how much a determined partition could capture the general trend contained in the base partitions. That is, if a partition is similar to the set that it is contained, that partition is of high quality. Diversity was defined as the opposite: the more a partition is different from all the others, the more diverse it is.

Considering these two definitions, the selection works as follows. First, the partition with the highest quality in the base partitions is selected. Next, it iteratively selects, from the base partitions, the most diverse partition compared to those selected in previous iterations, until the desired number of partitions is obtained.

3.3.4. Automatic Selection Algorithm (ASA)

ASA [Sakata et al. 2010], as the other CES methods, aims to select a reduced and diverse set of partitions, given an initial collection. ASA can be summarized in two main steps. First, it considers as part of its result the most evident partitions, that is, the ones that were repetitively found in the collection. The second step consists of an iterative search for a partition π_i with the highest similarity among the partitions not yet selected. When π_i is found, all others that are similar to it are discarded and π_i is added to the selected set. This iterative process continues until a certain reduction threshold is reached. ASA can be executed online at <http://lasid.sor.ufscar.br/asaproject/>.

3.4. MOCLE

MOCLE, as any cluster ensemble, is composed of two main steps: (i) generation of a diverse set of base partitions and (ii) determination of a consensus clustering solution [Faceli et al. 2009]. MOCLE differs from a typical cluster ensemble technique in two ways. First, it looks for a set of consensus partitions, instead of only one. This set of solutions may contain partitions that are combinations of other partitions or high-quality partitions from the initial set of base partitions. Second, it combines, iteratively, pairs of partitions in an optimization process, instead of the usual combination of all partitions at the same time. An advantage of this iterative combination of partitions is that it avoids possible negative influences from low-quality base partitions, which can decrease the quality of the ensemble results.

3.5. Experiments' Setup

As briefly pointed out before and shown on Figure 1, each of the chosen selection strategies received the pool of 35 partitions generated as input, producing the corresponding reduced set. These reduced sets were given as base partitions to MOCLE, resulting in several sets of outputs. Moreover, as a baseline to the analysis performed, MOCLE was run with the 35 initial pool of partitions, without any selection method. This baseline is called *full pool*, from now on.

Except for ASA, all the other selections require as input the number of clusterings to be contained in the reduced set. Since ASA is the only selection that automatically determines the size of the reduced set, in our first experiment, called Experiment 1 henceforth, we chose to use the same number of partitions selected by ASA as parameter for the other ones. ASA's output, following the parameterization described in [Sakata et al. 2010], contained 19 partitions, which corresponds to approximately 50% of the size of the initial pool. Therefore, in Experiment 1 we set, for all selections, a reduction to 19 partitions to be used as base partitions by MOCLE.

We also carried out a second experiment, called Experiment 2. ASA was left out from this experiment, as it cannot receive as parameter the size of the reduced set. All the other selections were tested aiming a 75% reduction in the partitions' pool, that is, 25% of the number contained in the full pool, 9 partitions. The intent of Experiment 2 was to analyze whether a further reduction is as effective or problematic as Experiment 1.

3.6. Evaluation of Results' Quality

In the present work, we use the Adjusted Rand Index (ARI) [Hubert and Arabie 1985] to evaluate the quality of MOCLE's results according to the underlying known structures present in the `golub` dataset (k2 and k3). ARI compares two partitions and it results in a statistic that varies in the interval $[-1, 1]$, where 1 indicates a perfect agreement between the partitions, while values less than or close to 0 correspond to an agreement found by chance.

The selection's impact analysis on the partitions produced by MOCLE was performed in two ways. First, just considering the best partitions in comparison to each known partition, and second in all the other clusterings also present in MOCLE's result sets. For the impact on the best partitions, we select the most similar partition to each known structure, per selection, per run. Then, we calculate the ARI's average of these with respect to the most similar known structure. For the impact on all the partitions, we consider all of them, per selection, per run, and calculate the ARI's average with respect to k2 and k3. Additionally, we also evaluate all the partitions contained in the initial pool.

The intention of these analysis is to observe if the selections would improve or not the partitions outputted by MOCLE with respect to the known structures.

4. Results

Experiment's 1 and Experiment's 2 averages (and standard deviations) of the best partitions, calculated as defined in Section 3.6, are listed in Table 1 and Table 2, respectively. The first two rows of these tables list the baselines for our experiments. The first one, initial pool, lists the ARI of the best partitions contained in the initial with respect to k2

Table 1. Impact of Experiment 1 on the best partitions with respect to k2 and k3.

Type	Selection	Average of the best		Number of Partitions Outputted (Average)
		k2	k3	
Baseline	Initial Pool	0.9439	0.8651	35
	Full Pool	0.8760 (0.00)	0.8651 (0.00)	12
SIS	Dunn	0.0299 (0.00)	-0.0049 (0.00)	6
	SS	0.7321 (0.25)	0.5917 (0.12)	9.3
	ASS	0.8958 (0.15)	0.6906 (0.07)	11.3
	VRC	0.9012 (0.05)	0.8839 (0.03)	12.4
	DB	0.6550 (0.15)	0.8016 (0.02)	8.4
	PBM	0.9126 (0.04)	0.8902 (0.03)	12.7
CRI	SR	0.6863 (0.27)	0.5687 (0.13)	9.4
	BRP	0.9204 (0.13)	0.6727 (0.05)	7.4
	SRD	0.9741 (0.03)	0.8765 (0.03)	11.8
Others	Diversity	0.7932 (0.00)	0.5551 (0.00)	13
	ASA	0.8783 (0.01)	0.7981 (0.00)	10.3

and k3. The second one, full pool, lists the ARI of the best partition outputted by MOCLE using initial pool as input, that is, MOCLE's result without any selection method.

Experiment 1 (Table 1) shows that the use of any selection method decreases the number of partitions outputted by MOCLE in about 15%, with respect to the full pool. This reduction is important, because it facilitates posterior manual analysis by domain experts [Sakata et al. 2010]. Moreover, Table 1 shows that, for k2, three of the SIS methods (Dunn, SS and DB) had a significant negative impact on MOCLE's results in comparison with the baselines, while the other three (PBM, VRC and ASS) improved the results. This unstable behavior was, in some degree, expected, because as [Naldi et al. 2013] pointed out, the SIS methods are prone to be biased towards its definition of quality, and when this definition does not match the true data model in question, poor results are expected. Dunn is an example of this mismatch, as it only selected very low quality partitions from the initial pool with respect to k2 and k3 ($ARI < 0.02$). This indicates that Dunn criteria do not match golub's known data model, at least not as well as the others relative quality indexes. Furthermore, MOCLE's results for k3 were also affected by the inconsistencies of the SIS methods. In summary, these observations strengthen the hypothesis that SIS methods are unreliable, including PBM and VRC — that showed excellent results —, as they are susceptible to be biased and might produce bad results like those showed by Dunn.

Table 1 also shows, regardless the SIS methods, that quality alone was not suffice to achieve good performance in MOCLE's results. BRP and SR are examples of this, as their focus is mainly in quality. Among the CRI methods only SRD, that considers both quality and diversity, led to improvements in finding the known partitions (k2 and k3). SRD helped MOCLE to produce better partitions than those contained in the initial

Table 2. Impact of Experiment 2 on the best partitions with respect to k2 and k3.

Type	Selection	Average of the best		Number of Partitions Outputted (Average)
		k2	k3	
Baseline	Initial Pool	0.9439	0.8651	35
	Full Pool	0.8760 (0.00)	0.8651 (0.00)	12
SIS	Dunn	0.0104 (0.00)	-0.0049 (0.00)	3
	SS	0.9440 (0.00)	0.6827 (0.00)	3
	ASS	0.3423 (0.00)	0.4139 (0.00)	3.1
	VRC	0.9863 (0.02)	0.8110 (0.03)	8.7
	DB	-0.0129 (0.00)	-0.0108 (0.00)	3
	PBM	0.8064 (0.28)	0.6242 (0.12)	8
CRI	SR	0.3615 (0.06)	0.4139 (0.00)	4.1
	BRP	0.9440 (0.00)	0.6827 (0.00)	4.4
	SRD	0.8793 (0.17)	0.6779 (0.08)	9.5
Others	Diversity	0.7451 (0.31)	0.6075 (0.14)	9.8
	ASA	–	–	–

pool in all of its executions, and in 17 of them helped to produce partitions equal to k2 (i.e., $ARI = 1$), something that MOCLE alone could not do. It also, in 3 of MOCLE's executions, helped to produce very similar partitions to k3 ($ARI = 0.94$). On the other hand, ASA and Diversity, which focus mainly in selecting a diverse set of partitions, did not result in improvements in comparison with the baselines. This reinforces the idea that both quality and diversity of the partitions must be considered in order to have a more significant positive impact on MOCLE's results.

Experiment 2 (Table 2), followed the trend shown in Experiment 1 and the use of any selection method decreased, in about 50%, the number of partitions outputted by MOCLE, in comparison with the full pool. However, Experiment 2 caused loss of performance in several selection methods, in comparison with Experiment 1. Only for SS, a further reduction in the number of partitions was beneficial in both cases (k2 and k3). For all the other selections, the reduction was harmful or did not make a difference in MOCLE's results, with respect to one or both cases, especially for DB that produced poor results like Dunn. SRD, that produced good results in Experiment 1, also showed loss in Experiment 2, but remained somewhat stable in comparison with SR. A possible explanation for SRD loss, is that the reduction in the number of partitions fed to MOCLE hurt its ability to produce good clusterings. Diversity also remained stable. These observations show that a further reduction, at least for this set of partitions and dataset, had a negative impact on MOCLE's results considering k2 and k3. However, this loss came with the advantage of a lower number of partitions outputted.

In both experiments, SRD and BRP were most successful methods, as they had a positive impact on MOCLE. In most cases, performing significantly better than MOCLE alone and helping it to find partitions not contained in the initial pool. Another positive

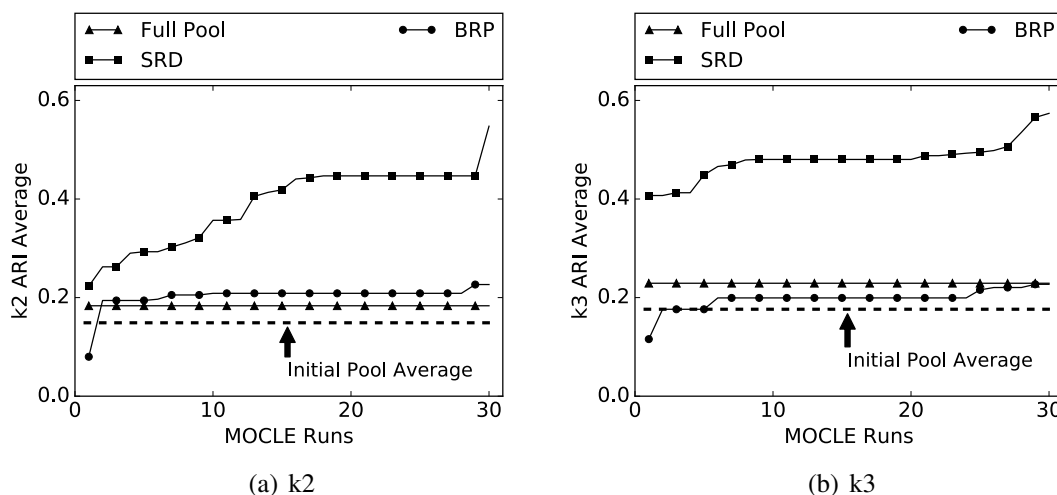


Figure 2. Experiment's 1 best selections ARI averages with respect to each known partition.

factor was the stability of the both methods, although more experiments are necessary to better evaluate this characteristic.

We also analyzed the impact of selection on the whole MOCLE's output of Experiment 1. The ARI's averages, calculated as defined in Section 3.6, with respect to k2 and k3 were plotted on the Figure 2(a) and Figure 2(b), respectively. We show only the executions of the best methods. The averages of each run were sorted for better visualization.

In both plots, it is possible to note that the selections resulted in a greater variability than the full pool and that for SRD even the selections' worst results performed better than the baselines (full pool and initial pool). For BRP, improvements were obtained in most cases for k2, but not for k3. This reinforces the hypothesis that these selections can indeed contribute to MOCLE's framework, especially when quality and diversity are both considered, increasing the overall quality of the outputted partitions.

Another positive impact of the selections was the effort reduction in MOCLE's processing. In our experiments, a significant reduction in MOCLE's runtime was noted due to the reduction in the MOCLE's base partitions number. In comparison with the full pool, Experiment 1 had its runtime decreased by around 50%. In Experiment 2, a 75% reduction was observed. This reduction of effort, although not yet fully sustained by more experiments, was important, because one of MOCLE's most prohibiting aspects is its long runtime as pointed out in [Faceli et al. 2009].

5. Conclusion and Future Work

We conducted preliminary experiments to evaluate the potential of quality improvement on MOCLE's results — with a reduced number of selections and using a well-known dataset — and identified that MOCLE can indeed benefit from partition selection, as other algorithms from ensemble and multiple clustering. These experiments also showed a reduction in the number of partitions outputted by MOCLE — facilitating posterior manual analysis — and improvements in its runtime. Our evaluation also reinforced other works

findings, as the unstable behavior of the SIS methods. We also observed that both quality and diversity in MOCLE's base partitions were important factors to produce good results, as in ensemble and multiple clustering. In future investigations, besides expanding the number of datasets, we also intend to explore different trade-offs between the number of partitions fed to MOCLE, its runtime and the final quality and stability of the results. Additionally, we plan to investigate more selection methods, that were left out of our experiments, including those successfully applied in others subareas of multiple clustering, as in meta-clustering, for example [Lei et al. 2014, Lei et al. 2016].

Acknowledgments

This work is funded by Coordination for the Improvement of Higher Education Personnel (CAPES) and the grant #2015/21560-0 of the São Paulo Research Foundation (FAPESP).

References

- Azimi, J. and Fern, X. (2009). Adaptive Cluster Ensemble Selection. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI'09*, pages 992–997, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Bailey, J. (2013). Alternative Clustering Analysis: A Review. In Aggarwal, C. C. and Reddy, C. K., editors, *Data Clustering: Algorithms and Applications*, pages 533–548. Chapman & Hall/CRC, 1st edition.
- De Hoon, M., Imoto, S., Nolan, J., and Miyano, S. (2004). Open Source Clustering Software. *Bioinformatics*, 20(9):1453–1454.
- Dudoit, S., Fridlyand, J., and Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457):77–87.
- Ertoz, L., Steinbach, M., and Kumar, V. (2002). A new shared nearest neighbor clustering algorithm and its applications. In *Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining*, pages 105–115.
- Faceli, K., de Souto, M. C. P., de Araújo, D. S. A., and Carvalho, A. C. P. L. F. (2009). Multi-objective clustering ensemble for gene expression data analysis. *Neurocomputing*, 72:2763–2774.
- Faceli, K., Sakata, T. C., de Souto, M. C. P., and de Carvalho, A. C. P. L. F. (2010). Partitions Selection Strategy for Set of Clustering Solutions. *Neurocomput.*, 73(16-18):2809–2819.
- Fern, X. Z. and Lin, W. (2008). Cluster ensemble selection. *Statistical Analysis and Data Mining*, 1(3):128–141.
- Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., and others (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537.
- Hadjitodorov, S. T., Kuncheva, L. I., and Todorova, L. P. (2006). Moderate Diversity for Better Cluster Ensembles. *Inf. Fusion*, 7(3):264–275.

- Hubert, L. J. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.
- Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Kuncheva, L. I. and Hadjitodorov, S. T. (2004). Using diversity in cluster ensembles. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, volume 2, pages 1214–1219 vol.2.
- Lei, Y., Vinh, N. X., Chan, J., and Bailey, J. (2014). FILTA: Better View Discovery from Collections of Clusterings via Filtering. In Calders, T., Esposito, F., Hüllermeier, E., and Meo, R., editors, *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II*, pages 145–160. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Lei, Y., Vinh, N. X., Chan, J., and Bailey, J. (2016). rFILTA: relevant and nonredundant view discovery from collections of clusterings via filtering and ranking. *Knowledge and Information Systems*, pages 1–41.
- Liu, L. M. and Fan, X. P. (2012). A Survey: Clustering Ensemble Selection. In *Advanced Materials Research*, volume 403, pages 2760–2763. Trans Tech Publ.
- Muller, E., Gunnemann, S., Farber, I., and Seidl, T. (2010). Discovering Multiple Clustering Solutions: Grouping Objects in Different Views of the Data. In *Proceedings of the 2010 IEEE International Conference on Data Mining, ICDM '10*, pages 1220–, Washington, DC, USA. IEEE Computer Society.
- Naldi, M. C., de Carvalho, A. C. P. L. F., and Campello, R. J. G. B. (2013). Cluster Ensemble Selection Based on Relative Validity Indexes. *Data Min. Knowl. Discov.*, 27(2):259–289.
- Rastin, P. and Kanawati, R. (2015). A Multiplex-network Based Approach for Clustering Ensemble Selection. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, ASONAM '15*, pages 1332–1339, New York, NY, USA. ACM.
- Sakata, T. C., Faceli, K., de Souto, M. C. P., and de Carvalho, A. C. P. L. F. (2010). Improvements in the Partitions Selection Strategy for Set of Clustering Solutions. In *Proceedings of the 2010 Eleventh Brazilian Symposium on Neural Networks, SBRN '10*, pages 49–54, Washington, DC, USA. IEEE Computer Society.
- Strehl, A. and Ghosh, J. (2003). Cluster Ensembles — a Knowledge Reuse Framework for Combining Multiple Partitions. *J. Mach. Learn. Res.*, 3:583–617.
- Vendramin, L., Campello, R. J. G. B., and Hruschka, E. R. (2010). Relative Clustering Validity Criteria: A Comparative Overview. *Stat. Anal. Data Min.*, 3(4):209–235.
- Zimek, A. and Vreeken, J. (2015). The Blind Men and the Elephant: On Meeting the Problem of Multiple Truths in Data from Clustering and Pattern Mining Perspectives. *Mach. Learn.*, 98(1-2):121–155.

Análise de dados obtidos através de um sistema de telemetria automotivo utilizando K-NN

Wilson Borba da Rocha Neto¹, José Maria Pires de Menezes Jr.¹, Rhulio Victor Luz Souza¹

¹Departamento de Engenharia Elétrica, Universidade Federal do Piauí (UFPI) – Campus Universitário Ministro Petrônio Portella-Bairro Ininga – CEP 64049-550 – Teresina – PI- Brazil

wsmartpc13@gmail.com, josemenezesjr@ufpi.edu.br, rhuliovictor@gmail.com

Abstract. *This work presents the construction of a wireless embedded system for automotive monitoring capable of capturing data from the electromechanical system and informing the location of a vehicle in real time. This system was designed to describe the behavior of the car and its driver. This work proposes to use the information obtained with the OBD II diagnostic system, applying them in a K-NN classification algorithm, in order to evidence driving models of a car, classifying the profiles of the drivers so that it is possible to suggest changes in behavior and prevent mechanical failure.*

Resumo. *Este trabalho apresenta a construção de um sistema para monitoramento automotivo capaz de capturar dados do sistema eletromecânico e informar a localização de um veículo em tempo real. Tal sistema foi projetado com o intuito de descrever o comportamento do automóvel e de seu condutor. Desta forma, este trabalho propõe utilizar as informações obtidas com o sistema de diagnóstico OBD II, aplicando-as em um algoritmo de classificação K-NN, com o intuito de evidenciar modelos de condução de um automóvel, classificando os perfis dos motoristas para que seja possível sugerir mudanças de comportamento e prevenir falhas mecânicas.*

1. Introdução

A telemetria é um sistema eletrônico de monitoramento construído para rastrear, medir ou enviar comandos a algum equipamento que esteja distante ou em constante movimento. Etimologicamente, a palavra telemetria tem origem a partir do grego *tele*, que significa remoto, e *metron*, que quer dizer medida. Ou seja, a telemetria pode ser entendida como a medição, processamento de dados e tratamento de informações a distância. O monitoramento, em tempo real, das funções eletromecânicas de um veículo tornou-se possível graças ao avanço da eletrônica automotiva, iniciado pela Bosch, em 1917, com o desenvolvimento da injeção eletrônica [Bosch 2011].

Atualmente, a complexidade da rede eletrônica automotiva já é tão grande que, do ponto de vista da ciência da computação, o carro é um sistema em tempo real, complexo e distribuído, com até 60 microcontroladores baseados em *Electronic Central Unit* (ECU) que se comunicam via, aproximadamente, 2500 sinais de diferentes tipos [Steinbach, et al. 2011]. ECU é, na verdade, um termo que generaliza qualquer sistema que controle um sistema ou subsistema de um veículo.

Tal evolução da eletrônica veicular permite a leitura de um grande volume de informações, as quais podem ser acessadas através da interface *On Board Diagnostic*

(OBD), atualmente na versão OBD II. A TSBE (2004) define o OBD II como um padrão de *Data Link Conector* (DLC) com uma conexão e pinos especificados para cada função, protocolos eletrônicos, códigos de diagnósticos de problemas, lista de parâmetros a serem monitorados e armazenados e uma terminologia padrão.

Para se ter ideia, o OBD II é capaz de acessar informações instantâneas que envolvem velocidade, aceleração, rotações por minuto, posição do pé do acelerador, carga no motor, consumo de combustível, temperatura do motor, distância percorrida, tensão na bateria e ainda realizar o diagnóstico de problemas dos sistemas eletromecânicos do automóvel [Szalay, et al. 2015].

Dessa forma, a Telemetria auxilia nas ações voltadas para a manutenção, consumo de combustível e na redução de riscos de acidentes. Porém, ela sozinha, não toma ação nenhuma. Existe a necessidade de organizar dados, encontrar padrões, associações, mudanças e anomalias relevantes. Para tal é necessário um conjunto de ferramentas e técnicas que através do uso de algoritmos de aprendizagem ou classificadores baseados em redes neurais e estatísticas, são capazes de explorar um conjunto de dados, extraíndo ou ajudando a evidenciar padrões nestes dados e auxiliando na descoberta de conhecimento.

Pode-se identificar um motorista com um perfil pouco cuidadoso, que tenha um comportamento agressivo no trânsito, coletando apenas, algumas informações do automóvel. Como exemplo, um motorista apressado ou não muito cuidadoso tem geralmente o perfil de freadas bruscas. Observar a frequência e intensidade das freadas já seria suficiente para a diferenciação de tipos de condução, mas dificilmente um condutor tem um comportamento ruim observado apenas uma única variável.

Para montar padrões e classificar perfil de condução faz-se necessário então um conjunto de observações. Quanto mais informações forem coletadas, melhor será a descoberta de conhecimento, sendo assim, por exemplo, analisar a intensidade das acelerações também seria importante para reconhecer um comportamento agressivo no trânsito.

Dada a importância do monitoramento, do diagnóstico e da evolução dos sistemas ao longo do tempo, o objetivo deste estudo é desenvolver um sistema embarcado micro-controlado integrado ao automóvel, capaz de conectar-se a um servidor web, utilizando a rede de telefonia móvel para o envio de informações. Este possibilita a criação de um banco de dados online com as informações obtidas do OBD II, os quais serão utilizados no algoritmo de reconhecimento de padrões.

O algoritmo utilizado é o classificador clássico conhecido como o K vizinhos mais próximos (do inglês: *K nearest neighbors* – K-NN), com o intuito de encontrar padrões, associações, mudanças e anomalias relevantes. Com isso, o sistema projetado, e descrito neste trabalho, é capaz de classificar e evidenciar padrões, para que seja possível identificar a atual situação do automóvel e seu motorista, avaliar a condução, criar um banco de dados de funcionamento do veículo, instruir correções de comportamento e prever problemas mecânicos.

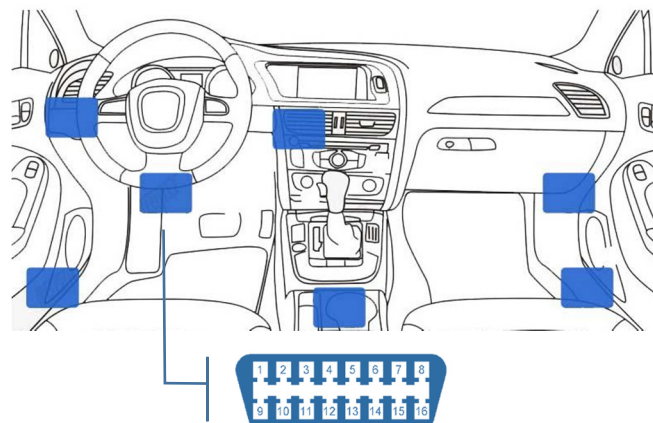
No tocante a construção do artigo, a seção 2 apresenta uma descrição detalhada do sistema OBD II. Uma breve explicação do sistema projetado para captura é abordada na seção 3. A seção 4 aborda a definição do classificador utilizado, introduzindo conceitos

que facilitam a compreensão do proposto na seção 5, a qual apresenta os resultados alcançados. Conclusão e trabalhos futuros, trazidos na seção 6, fecham o documento.

2. OBD II

Para que se possa receber informações do veículo, este deve possuir uma rede de comunicação interna capaz de monitorar seus sistemas eletromecânicos e entender as requisições realizadas por um agente externo. Uma dessas é a *Controller Area Network* (CAN), com protocolo de padronizado pela norma ISO 15765-2, que, tal qual explicado por Barbosa (2003), busca a simplificação da rede eletrônica de um automóvel, utilizando-se de um sistema distribuído, composto por vários nós de interconexão e multimestre.

As especificações do OBD II incluem a padronização do conector e da função de seus pinos. Como é informado na norma SAE J1962, o conector, que deve estar posicionado até 0,61m do volante, é composto por duas colunas de oito pinos, como pode ser visto na Figura 1. Esta também exemplifica a função de cada pino presente na camada física do sistema, evidenciando que os pinos possuem diferentes papéis dependendo do protocolo utilizado e de opções implementadas pelo fabricante.



Pino	Descrição	Pino	Descrição
1	Opção do vendedor	9	Opção do vendedor
2	SAE J1850 Bus+	10	SAE J1850 Bus-
3	Opção do vendedor	11	Opção do vendedor
4	Terra do Chassis	12	Opção do vendedor
5	Terra do Sinal	13	Opção do vendedor
6	CAN (ISO 15765-4 e SAE J2284) High	14	CAN ISO (15765-4 e SAE J2284) Low
7	ISO 9141-2 K-Line	15	ISO 9141-2 L-Line
8	Opção do vendedor	16	Bateria do automóvel

Figura 1. Função dos pinos presentes no conector OBDII e sua localização

Além do controle interno dos sistemas do automóvel ser capaz de identificar as requisições externas é necessário um sistema que possa enviá-las pelo terminal. Para

requerer quaisquer dados, o dispositivo compatível conectado ao terminal OBDII deve enviar comandos de diagnóstico ou parâmetros de informações, conhecidos como *Parameter Identification* (PID), padronizados pela SAE J1979 e ISO15031-5.

Os PIDs são divididos em um total de 10 serviços. Cada um possui uma série de códigos que podem ser utilizados para atividades específicas [SAE J1979 ,2012]. Neste trabalho, utiliza-se majoritariamente o serviço \$01, que contém os PIDs relacionados com as informações de funcionamento. Como o suporte a diferentes parâmetros depende do fabricante, também é disponibilizado, neste serviço, um responsável pela identificação dos PIDs implementados.

Como informado na SAE J1979, os hexadecimais que compõem cada parâmetro e sua resposta são uma forma de codificar as informações enviadas. Dessa forma, no PID **010D**, por exemplo, **01** indica o serviço ao qual o parâmetro pertence e **0D** a função específica do serviço escolhido. Assim, **010D** é um parâmetro que realiza uma captura da velocidade do veículo em tempo real. Essa identificação é ilustrada na Figura 2

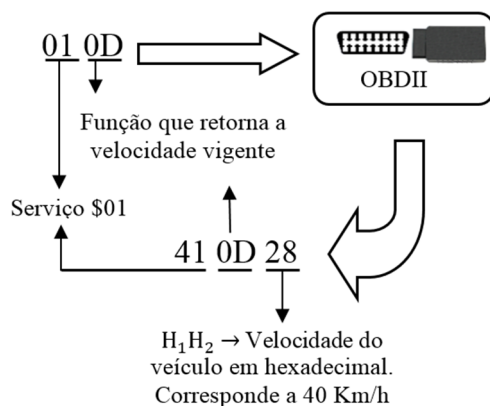


Figura 2. Exemplo de comunicação utilizando um PID

As respostas obtidas para cada PID, extraídas do sistema OBD II, são sequências de hexadecimais, as quais mudam para cada um dos parâmetros. A Figura 2 e a Tabela 1 mostram as formas de cálculo para as informações coletadas e utilizadas neste trabalho. Note que os símbolos H₁, H₂, H₃ e H₄ representam a sequência de hexadecimais recebida como resposta útil para cada tipo de requisição.

Tabela 1. Parâmetros chave escolhidos para captura

PID	Informação	Cálculo
010D	Velocidade (Km/h)	H ₁ H ₂
010C	Rotações por minuto (RPM)	$\frac{(256 \times H_1H_2 + H_3H_4)}{4}$
0110	Mass Air Flow(MAF)(g/s)	$\frac{(256 \times H_1H_2 + H_3H_4)}{100}$
0111	Posição do pedal de aceleração (%)	$\frac{100}{(255)} \times H_1H_2$
0104	Carga no motor (%)	$\frac{100}{(255)} \times H_1H_2$

As informações chave para captura, ilustradas na Tabela 1, foram selecionadas com o intuito de garantir uma avaliação da situação do automóvel e do perfil de condução do motorista, pois são capazes de modelar o funcionamento do automóvel. Além disso, tais dados podem ser utilizados para o cálculo de informações não disponibilizadas pelo OBD II, como o consumo de combustível e tempo de parada, as quais serão obtidas por meio de fórmulas próprias, abordadas em seções posteriores.

3. Sistema de Telemetria

Para capturar e salvar as informações em um servidor utilizando o OBD implementou-se um sistema de telemetria embarcado no veículo. Este é capaz de organizar em um banco de dados online todas as informações anteriormente destacadas. Além disso, o uso do GPS permite a identificação e localização das capturas, o que adiciona ainda mais variáveis ao processamento.

Apesar de já existir opções no mercado que supram essa necessidade, optou-se por projetar um sistema próprio, já que o controle do fluxo de dados e modificações do sistema se dão de forma livre em servidores e protótipos próprios. Além disso, a maioria dos produtos existentes acoplam em uma mesma unidade física o sistema de captura e de envio, o que pode causar problemas em situações de roubo, no qual uma pessoa mal-intencionada poderia retirar o dispositivo e impedir o rastreamento do veículo.

Por conta disso, optou-se por desvincular fisicamente o envio da captura através do uso de uma conexão Bluetooth. O esquemático do sistema, que permite a visualização de suas funções e componentes, pode ser visto na Figura 3, esta também exemplifica o vetor de dados que é enviado ao servidor em cada captura.

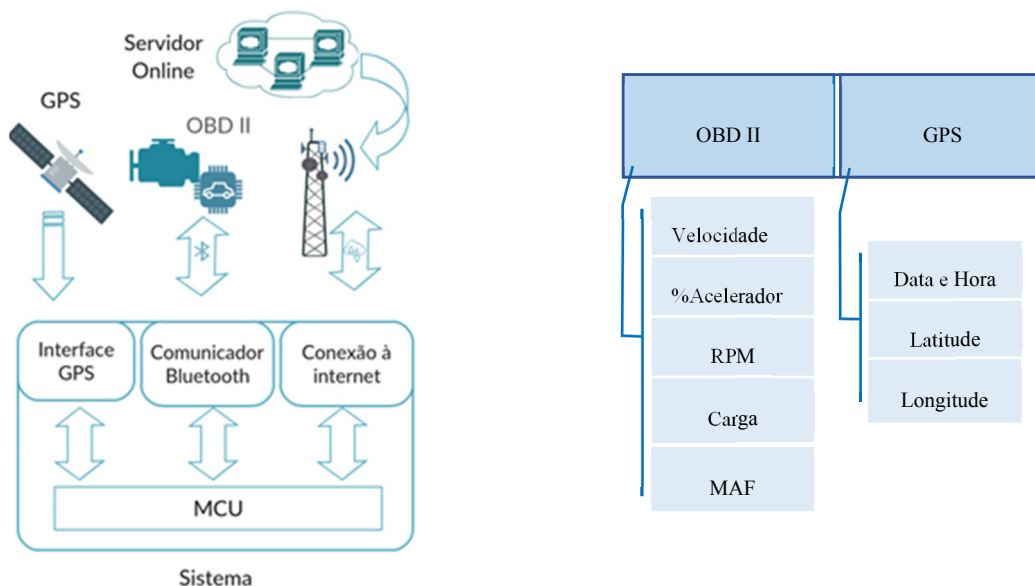


Figura 3. Sistema de telemetria automotiva e o vetor de dados enviado

Toda a informação enviada ao servidor é disponibilizada no banco de dados, para posterior processamento e implementação dos algoritmos de classificação. Além disso, para uma análise mais didática, as informações são disponibilizadas em gráficos e tabelas em um endereço eletrônico, permitindo que o usuário do sistema acompanhe as estatísticas das capturas realizadas em seu veículo.

3.1 Organização das informações capturadas

As informações disponibilizadas pelo servidor, após capturadas do veículo, são 5: rotações por minuto, fluxo de massa de combustível (MAF), velocidade, aceleração e carga no motor. Destas, acrescentou-se o cálculo do consumo de combustível com uma fórmula experimental (Eq. 1) baseada na análise dimensional das variáveis e de correções realizadas por testes empíricos.

$$\text{Consumo} \left(\frac{\text{L}}{\text{Km}} \right) = \frac{3600 \times \text{MAF}}{(\text{Razão}_{\text{COMBUSTÍVEL}}^{\text{AR}}) \times \text{Densidade} \times \text{Velocidade}} \quad (1)$$

Tais informações, chamadas de atributos na classificação, são agrupadas pelo servidor sequencialmente em arquivos de texto, coletadas a cada 0,5s. Os arquivos de cada veículo são salvos separadamente e identificados pelo número de série do sistema de captura correspondente.

4. Classificação de Padrões

O reconhecimento de padrão envolve, essencialmente, as seguintes etapas de processamento: pré-processamento, extração de parâmetros e classificação. O pré-processamento, tem o objetivo de eliminar dados desnecessários fazendo com que a entrada apresente apenas dados relevantes para o reconhecimento do objeto em análise.

A extração de parâmetros, também denominada etapa de seleção de atributos, consiste em análise dos dados de entrada a fim de extrair e derivar informações úteis para o processo de reconhecimento. O estágio final do reconhecimento de padrão é a classificação, ou etapa de tomada de decisão, que analisa as características extraídas dos dados de entrada e declara o objeto em análise como pertencente a uma determinada categoria [Matos 2008].

Classificações de padrões investiga o desenvolvimento de técnicas que permitem a construção de sistemas automáticos de aprendizado. Normalmente ele é dado em duas fases: treinamento, onde é aprendida uma descrição geral do conceito; e teste, que utiliza a descrição aprendida para categorização de novos dados passados ao sistema. Dentre os principais tipos de aprendizado pode-se citar o supervisionado, não-supervisionado e semi-supervisionado. No primeiro deles a classe de dados é fornecida por uma fonte externa ao processo de aprendizado, muitas vezes por um especialista humano. Já o aprendizado não-supervisionado utiliza a identificação de subconjuntos de dados que compartilham certas similaridades. Por fim, o aprendizado semi-supervisionado envolve um subconjunto de dados para induzir uma expressão geral do conceito e então determinam a classe dos dados restantes [Real 2014].

Assumindo que os dados podem ser representados como pontos no espaço euclidiano, eles podem ser organizados de acordo com um dos tipos de aprendizado. Dentre as técnicas mais comuns de classificação de padrões, pode-se destacar o algoritmo do K-Vizinhos mais Próximos (K-NN).

4.1 O Método de Classificação por K-NN

A abordagem supervisionada para classificação tem sido estudada extensivamente. Os métodos deste grupo supõem que características importantes estão inseridas nos vetores características. Então esses métodos procuram construir um mapeamento através de um banco de dados para serem utilizados por algoritmos de aprendizado de máquinas. Como um primeiro passo, o sistema é treinado com alguns dados conhecidos manualmente, e, em seguida ela é usada para classificar os dados não conhecidos [Scaringella, Zoia e Mlynek 2006].

O classificador de K-vizinhos mais próximos, proposto por Fukunaga (1975), é um classificador sub-ótimo que cria fronteiras de decisões complexas a partir de um conjunto de padrões de treinamento com classes conhecidas a priori [Matos 2008]. Mais precisamente, para um dado vetor característico no conjunto teste, os vetores de K mais próximos no conjunto de treino são selecionados (de acordo com alguma medida de distância, sendo que neste trabalho é utilizado a distância euclidiana) e o vetor de características de teste é atribuído ao rótulo mais representado na classe dos K vizinhos [Scaringella, Zoia e Mlynek 2006].

Classificar um exemplo desconhecido com o algoritmo K-NN pode ser resumido nos seguintes passos e visualizado na Figura 4:

- Calcular a distância entre o exemplo desconhecido e os outros exemplos do conjunto de treinamento;
- Identificar os K vizinhos mais próximos;
- Utilizar o rótulo da classe de vizinhos mais próximos para determinar o rótulo de classe do exemplo desconhecido em uma espécie de votação.

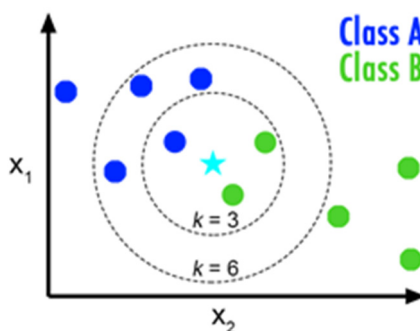


Figura 4. Ilustração da classificação de acordo com o K-NN

5. Resultados

5.1 Definição das classes

O intuito deste trabalho é projetar um sistema capaz de classificar o modo de condução de determinado veículo com a utilização do algoritmo de classificação K-NN. Por conta disso, inicialmente, definiram-se as classes que seriam utilizadas nesse processo. Optou-se pela utilização de três classes: Agressiva, Moderada e Leve.

Para compor o banco de dados o automóvel percorreu um circuito de teste controlado, isto é, um circuito onde foi possível preservar as mesmas características de via para cada modo de condução. Dessa forma, a cada percurso, uma forma de condução

diferente foi utilizada no automóvel. Ao fim, coletou-se uma quantidade igual de capturas para cada classe. Como uma forma de conhecer as classes utilizadas nesse projeto, optou-se por representar num gráfico de uma maneira bidimensional dois atributos coletados, que, nesse caso, foram RPM e Consumo. Assim, chegou-se ao gráfico apresentado na Figura 5, o qual possui a representação de pontos instantâneos de cada captura. Pode-se verificar uma certa separação entre as classes, mesmo utilizando apenas dois atributos.

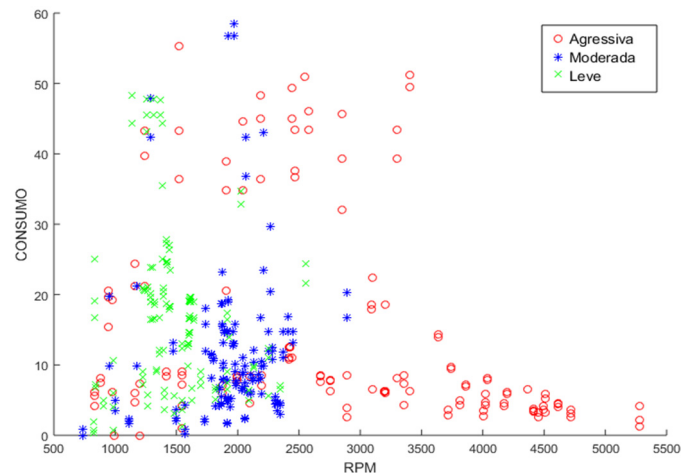


Figura 5. Dados de 2 atributos capturados divididos em 3 classes

Deve-se destacar que a separação entre as classes é de menor intensidade em pontos próximos à origem, RPM baixo. Essa região descreve condições de parada, passagem por quebra-molas e cruzamentos; comportamento esse, presente em todos os modos de condução. Porém essa região é mais densamente ocupada pela classe Leve, pois nesse modo de condução o veículo prioriza baixo RPM.

5.1 Algoritmo de classificação

Com as classes e atributos definidos, foram coletados dados após o carro percorrer o trajeto algumas vezes. Ao fim, foram feitas 9.000 capturas/amostras, distribuídas nas três classes descritas anteriormente: Agressiva, Moderada e Leve.

Para melhor preservar a informação temporal das amostras, os dados foram organizados em janelas temporais. Isto é, foram organizados blocos de l capturas, sequencialmente reunidos. Sendo assim, a nova amostra possui dimensão l vezes o número de atributos. Essa construção é esboçada na Figura 6.

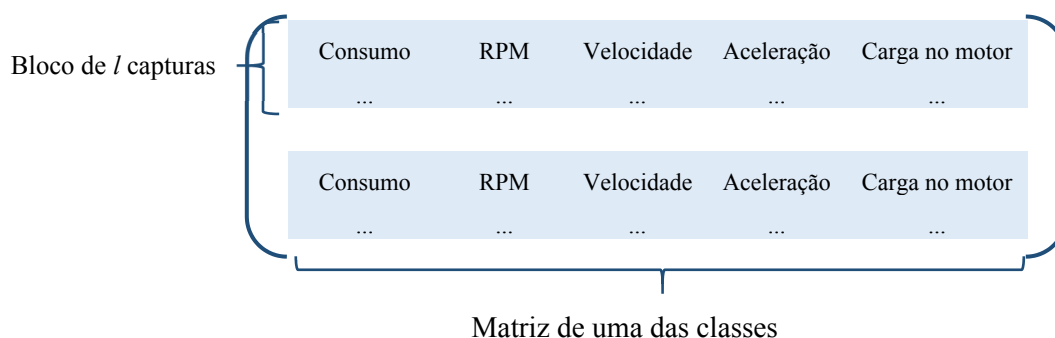


Figura 6. Matriz de dados para cada classe

O método de validação *holdout* foi escolhido para verificar o desempenho da classificação. Assim os blocos foram postos de forma aleatória e separados em matrizes de treino e matrizes de teste. Optou-se por uma divisão de 80%/20%, ou seja, os primeiros 80% das amostras foram utilizados para o treinamento e o restante foi utilizado para os testes. Esse processo é exemplificado na Figura 7.

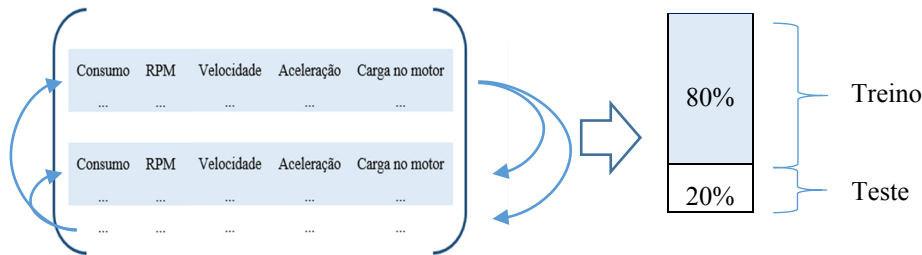


Figura 7. Proporção dos dados

Com a divisão em blocos de atributos, a classificação é efetuada comparando, individualmente, cada bloco de teste com os das classes de treino. Dessa forma, cada um dos blocos de determinada matriz de teste será classificado independentemente dos outros. Assim, para definição do perfil do condutor é selecionada a classe à qual foi atribuída a um maior número de blocos após efetuada a classificação. Caso ocorra empate no número de blocos indicados a cada classe, será escolhida aquela que possuir a maior proximidade dentre todos os cálculos efetuados. Tal situação é exemplificada na Figura 8.

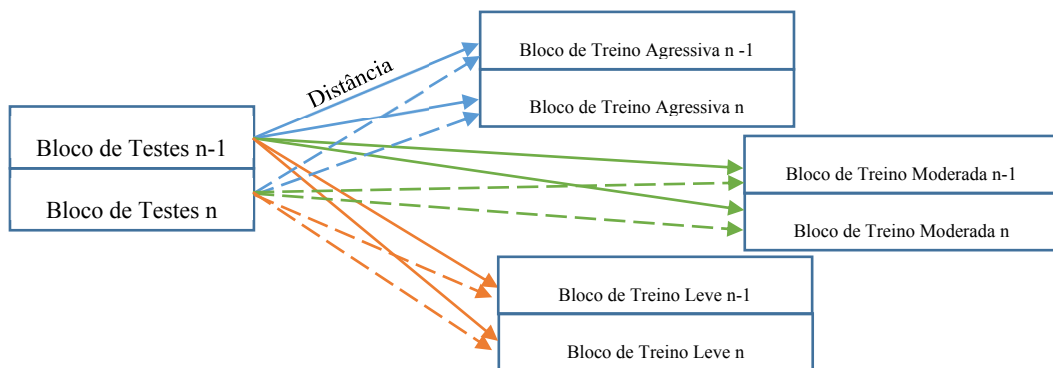


Figura 8. Exemplificando o cálculo da distância

5.2 Resultados da classificação

Cada rodada de treino/teste foi repetida 1.000 vezes, já que as matrizes são embaralhadas a cada rodada, permitindo diferentes resultados de classificação. Assim, obteve-se média, mediana e variância dos acertos e erros do classificador.

Primeiramente foram feitos testes de variação do tamanho da janela de capturas. O gráfico apresentado na Figura 9 mostra a importância da construção dessa janela. Pode-se verificar que blocos a partir de 5 capturas sequencialmente coletadas tendem a trazer melhor desempenho para o classificador. Isso se deve porque as características de condução têm influência temporal, por exemplo uma aceleração forte, uma freada brusca ou condições de parada.

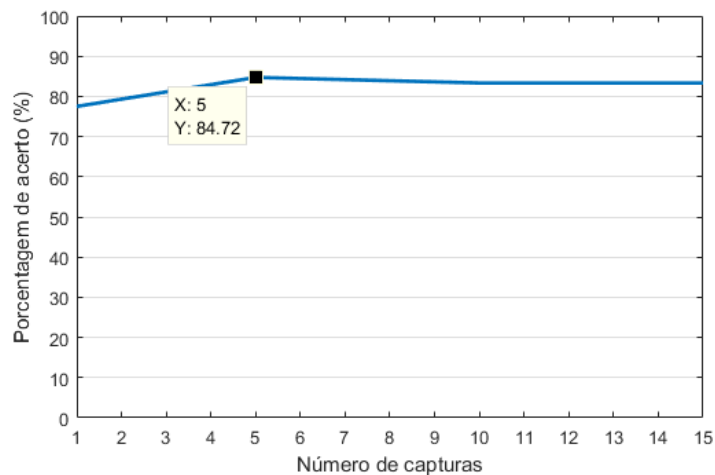


Figura 9. Variação do número de capturas na taxa de acerto da classificação

Outro ponto analisado foi o número de vizinhos mais próximos do classificador K-NN. O gráfico apresentado na Figura 10 mostra o efeito de tal variação na porcentagem de acerto da classificação. Pode-se verificar que a partir de 15 vizinhos o classificador já possui um desempenho razoável, não trazendo melhor taxa de classificação com o aumento do número de vizinhos. O valor máximo de taxa de acerto se deu com 21 vizinhos, aproximadamente 85% de taxa de acerto.

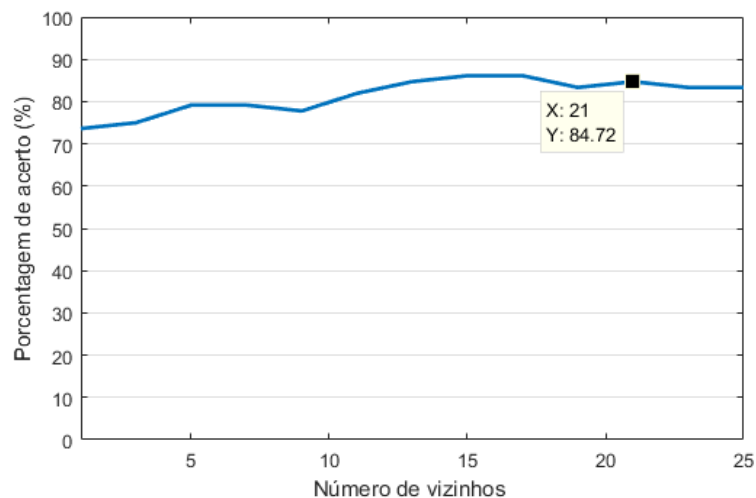


Figura 10. Variação do número de vizinhos na taxa de acerto da classificação

Com base nos testes, encontrou-se o ponto ótimo do classificador, ou seja, a quantidade de capturas e o número de vizinhos que permitiu ao classificador atingir um maior número de acertos. Este ponto consiste em 5 capturas por bloco e 21 vizinhos, com uma taxa de acerto de aproximadamente 85%. A matriz de confusão, neste ponto, é apresentada na Tabela 2, com porcentagem de erro e acerto para cada classe. Já a Tabela 3 descreve estatisticamente, através de média, mediana e variância, os resultados obtidos no ponto ótimo do classificador.

Observa-se com os dados da Tabela 2 que características da classe Lenta, como apontado pela Figura 6, estão presentes nas outras classes, o que é inerente da condução, já que qualquer motorista passa por situações de baixa velocidade e baixa rotação. Além disso, a classe Moderada apresentou o maior índice de erro, já que, por ser uma classe

intermediária, possui características de ambas as outras. A classe Lenta apresentou a maior taxa de acerto por possuir atributos bem mais concentrados, comportamento também visualizado na Figura 6.

A informação de média e de mediana dos acertos do processo de classificação estão bem próximas, o que indica uma distribuição centrada dos resultados obtidos. A existência de valores distantes da média, indicada pelo valor elevado da variância, não influenciaram de forma consistente as outras medidas estatísticas, o que também foi impactado pelo grande número de iterações.

Tabela 2. Matriz de confusão do classificador K-NN

		Classe Verdadeira		
		Agressiva	Moderada	Lenta
Classe Estimada	Agressiva	89,75%	12,51%	0%
	Moderada	0%	66,67%	0%
	Lenta	10,25%	20,82%	100%

Tabela 3. Resultados do classificador

Dados Estatísticos		
Média	Mediana	Variância
84,72	85,00	18,30

6. Conclusão

Sistemas de telemetria fornecem uma série de dados e informações sobre vários aspectos do funcionamento dos veículos e do comportamento do motorista aos gestores de uma frota. Esses dados e informações diversas podem ser transformados em indicadores importantes, que se aplicados corretamente na gestão de frotas, podem proporcionar uma série de benefícios.

Assim, esse sistema permite acompanhar a forma como os motoristas dirigem e identificar padrões de condutas desejados. Com isso esse sistema ajuda a reduzir os custos da frota, fazendo com que os motoristas dirijam de forma mais consciente, reduzindo a velocidade – e, por consequência, o consumo de combustível e a ocorrência de acidentes.

Por fim, deve-se destacar que o sistema permite guardar e analisar diversas variáveis geradas pelos veículos, possibilitando o emprego de técnicas de aprendizagem de máquinas para reconhecimento de padrões e predição de falhas ou defeitos no veículo, não somente no reconhecimento do comportamento do condutor.

7. Referências

BARBOSA, L. R. G. (2003). **Rede CAN**, Belo Horizonte: UFMG.

BARONE, S.; D'AMBROSIO, P.; ERTO, P. **A statical monitoring approach for automotive on-board diagnostical system.** Quality and international reliability engineering international. [S.l.]: [s.n.]. 2006. p. 555-575.

BOSCH. **Bosch Automotive: A product History.** Estugarda (Stuttgart): Bosch, 2011. pg. 41 -43 p.

CÂMARA, J.; CERQUEIRA, T. A.; DA SILVA, V. L. Development of an automotive scanner for educational application. **IEEE latin american transactions**, v. 15, nº 1, p. 40-47, 2017.

CHEN, S.-H.; WEI, Y.-R. A study on speech control inteface for vehicle on-board diagnostic system. **2010 Fourth International Conference on Genetic and Evolutionary Computing**, p. 614-617, 2010.

GODAVARTY, S.; BROYLES, S.; PARTEN, M. Interfacing to the On-board Diagnostic system. **Proceedings Vehicular Technology Conference** , v. 4, p. 24-28, 2000.

FUKUNAGA, K.; NARENDRA, P. M. **A branch and bound algorithm for computing k-nearest neighbors.** IEEE Transactions on Computers, v. 100, n. 7, p. 750–753, 1975

SCARINGELLA, Nicolas; ZOIA, Giorgio; MLYNEK, Daniel. **Automatic genre classification of music content: a survey.** Signal Processing Magazine, IEEE, v. 23, n. 2, p. 133-141, 2006

SOCIETY OF AUTOMOTIVE ENGINEERS. **SAEJ1962: Diagnostic Connector Equivalent to ISO/DIS 15031-3.** [S.l.]. 2001.

SOCIETY OF AUTOMOTIVE ENGINEERS. **SAEJ1979: E/E Diagnostic Test Modes.** [S.l.]. 2012.

STEINBACH, T.; KORF, F.; SCHMIDT, T. C. Real-time Ethernet for automotive applications: a solution for future in-car networks. **in Proceedings of the 1st IEEE International Conference on Consumer Electronics (ICCE '11)**, Berlin, Germany, p. 216–220, 2011.

SZALAY, Z. et al. ICT in road vehicles - reliable vehicle sensor information from OBD versus CAN. **2015 International Conference on Models and Technologies for Intelligent Transportation System (MT-ITS)**, p. 469-476, 2015.

REAL, E. M. **Investigação de algoritmos sequenciais de agrupamento com pré-processamento de dados em aprendizado de máquina.** Faculdade Campo Limpo Paulista. Campo Limpo Paulista. 2014.

MATOS, F. M. D. S. **Reconhecimento de faces utilizando a transformada cosseno discreta.** UFPB. João Pessoa, p. 94. 2008.

TRANSPORTATION SYSTEMS BRANCH ENVIRONMENT CANADA. **On-Board Diagnostics II (OBDII) and Light-Duty Vehicle Emission related Inspection (I/M) Programs**, p. 1-7, 2004.

D2DSS : Simulador Discreto 2D do Jogo de Futebol

Igor Oliveira Borges¹, Valdinei Freire¹ e Karina Valdivia Delgado¹

¹Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
São Paulo, SP – Brasil

{igor.borges, valdinei.freire, kvd}@usp.br

Abstract. *This article introduces a new simulator implemented in Octave/Matlab to provide a discrete mapping environment, that is easy-to-use and has a graphical engine separated from the execution module. This simulator allows to set up several environmental variables that allowed the simulation of the characteristics of a real game. Strategies were also implemented for agents in this simulator, based on static rules with A* search, random or with Reinforcement Learning techniques, Q-Learning and Sarsa. This project is available in an open repository to be used for both didactic and research purposes.*

Keywords. *Soccer Game Simulator, Reinforcement Learning, Heuristics, Markov Decision Processes.*

Resumo. *Este artigo apresenta um novo simulador implementado em Octave/Matlab, visando fornecer um ambiente de mapeamento discreto, de fácil utilização e que tenha a parte gráfica separada da execução. O simulador permite a configuração de diversas variáveis de ambiente que possibilitam a simulação das características de um jogo real. Ainda foram implementados estratégias para os agentes no simulador, baseada em regras estáticas com busca local A*, aleatória ou com as técnicas de Aprendizado por Reforço, Q-Learning e Sarsa. Disponibilizasse este projeto em um repositório aberto para ser utilizado tanto para fins didáticos quanto de pesquisa.*

Palavras-Chave. *Simulador do Jogo de Futebol, Aprendizado por Reforço, Heurísticas, Processos de Decisão Markoviano.*

1. Introdução

Simuladores de futebol são utilizados para o desenvolvimento de pesquisas especialmente ligadas à robótica. Porém, também são utilizados para desenvolver pesquisas em áreas como visão computacional, aprendizado de máquina, controle e automação, entre outros [Costa and Leite 2012]. Tais simuladores surgiram em conjunto com as competições de futebol de robôs, pela necessidade de simular o comportamento esperado dos robôs no jogo sem precisar do equipamento físico, reduzindo-se assim o custo de desenvolvimento.

A competição de futebol de robôs Robot J-League criada em 1993 no Japão é a precursora da competição internacional RoboCup, que ocorre desde 1997 anualmente sempre em um país diferente [Kitano et al. 1997]. Uma das modalidades dessa competição é a Simulation League [Gabel and Riedmiller 2011] que é uma categoria destinada a simulação do jogo de futebol utilizando o simulador 2D ou 3D da própria organização. Esses eventos estimulam que desenvolvedores elaborem e aprimorem algoritmos para melhorar a tomada de decisão no jogo de futebol usando diversas técnicas computacionais a fim de vencer os jogos simulados ou reais [Akiyama et al. 2014].

O simulador da RoboCup, bem como a maioria dos simuladores existentes usam um mapeamento contínuo e possuem muitas variáveis que demandam tempo para configurar o ambiente de trabalho, e mais ainda para que potenciais usuários possam de fato utilizá-lo em pesquisas [Gabel and Riedmiller 2011]. Além disso, simuladores disponíveis como o da Robot Soccer, SServer [Chen et al. 2003] e USPDS [Melo and Romero 2009] não são simples e não estão bem documentados [Costa and Leite 2012].

Neste artigo é apresentado um novo simulador do jogo de futebol chamado D2DSS (Discrete 2D Soccer Simulator). O simulador: (i) usa um mapeamento discreto; (ii) é simples pois tem menos variáveis para sua configuração; e (iii) separa a parte gráfica da execução, possibilitando que testes sejam realizados consecutivamente sem dependência da visualização. Essas características permitem que novos algoritmos de controle possam ser desenvolvidos no simulador em menos tempo, sem a necessidade de tratar o problema de localização dos jogadores.

O simulador proposto permite ainda escolher entre algumas estratégias pré-implementadas para serem utilizadas por cada time. Dentre as estratégias implementadas estão: uma estratégia aleatória, uma estratégia baseada em regras e mais duas estratégias que usam algoritmos de aprendizado por reforço (Q-Learning e Sarsa). Finalmente, foram realizados experimentos comparando o desempenho dessas estratégias.

O restante deste texto está organizado da seguinte forma. A Seção 2 descreve as características do simulador proposto. A Seção 3 apresenta as estratégias pré-implementadas no simulador. Na Seção 4 é comparado o desempenho dessas estratégias. Por fim, a Seção 5 apresenta a conclusão.

2. Simulador D2DSS

O simulador D2DSS¹ pode ser utilizado tanto para fins didáticos quanto de pesquisa. O simulador foi desenvolvido em Octave/Matlab e é compatível com diversos sistemas operacionais (Windows, Mac OS X, e Linux). Na Figura 1 é apresentada a interface gráfica do simulador. Os jogadores são representados por retângulos enumerados e a bola é representada por um quadrado pequeno branco.

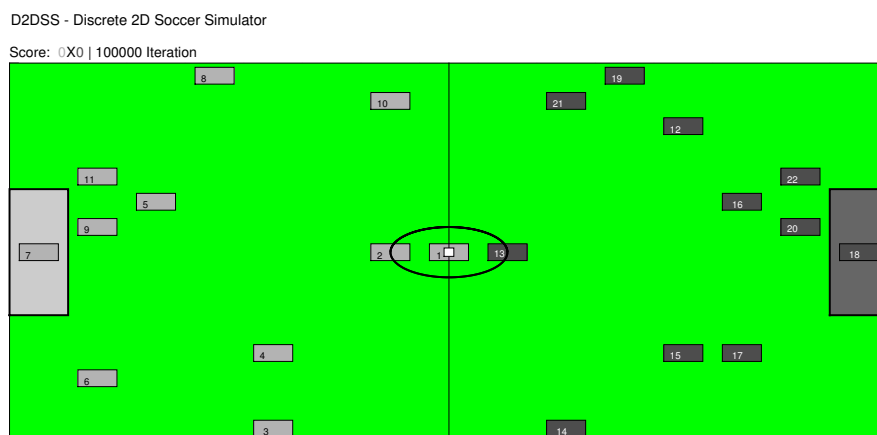


Figura 1. Interface gráfica do simulador D2DSS com onze jogadores por time em um campo 15×15 e gol 5×1 .

¹Disponível em: <https://github.com/D2DSS/Sim>

A seguir são descritos a modelagem do problema como um processo de decisão markoviano e a dinâmica do simulador; como é feito o controle da partida; os módulos implementados; e a configuração do simulador.

2.1. Modelagem e Dinâmica

O simulador D2DSS foi modelado como um processo de decisão markoviano (*Markov Decision Process* - MDP) e estruturado de forma a capturar as particularidades do jogo de futebol. Considera-se que o ambiente é totalmente observável, i.e., os agentes (jogadores) tomam decisões com base no estado atual do processo que é completamente conhecido.

Um MDP é definido pela tupla $\langle \mathcal{S}, \mathcal{A}, \mathcal{T} \text{ e } \mathcal{R} \rangle$, em que: \mathcal{S} é o conjunto de estados pertencentes ao processo; \mathcal{A} é o conjunto de ações que podem ser executadas durante episódios de decisão; $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ é a função que define a probabilidade de transição de estados no sistema; e \mathcal{R} é a função recompensa que define o resultado de tomar uma ação $a \in \mathcal{A}$ em um estado $s \in \mathcal{S}$ [Puterman 1994].

2.1.1. Estados

O estado é representado pelas coordenadas (x_i, y_i) de cada um dos n_P jogadores em campo, as coordenadas da bola (x_b, y_b) , e as variáveis de direção e velocidade da bola (d_b, v_b) . Assim, a quantidade de estados do problema está diretamente relacionada com o número de jogadores em campo (n_P) e também com o tamanho do campo.

2.1.2. Ações e Transições

Cada jogador pode escolher por rodada uma das 14 ações definidas. Essas ações foram divididas em 4 classes (*Parada*, *Movimento*, *Chute* e *Recuperação*) e são apresentadas na Tabela 1.

Tabela 1. Classificação das ações.

Id	Classe de ação	Restrição	Ações
1	Parada	Nenhuma	Stop
2	Movimento	Nenhuma	Move N, S, L, W
3	Chute	Posse da bola	Kick N, S, L, W, NE, SE, NW, SW
4	Recuperação	Adjacente à bola	Tackle

As ações da classe *Parada* e *Movimento* não possuem nenhuma restrição, i.e., o jogador pode ter a posse ou não da bola. Já as ações da classe *Chute* necessitam que o jogador tenha a posse da bola para sua execução, e por fim a ação da classe *Recuperação* necessita que o jogador esteja ao redor da bola, i.e., em alguma das 4 posições em volta de (x_b, y_b) ; se uma das ações executadas não atende a restrição, então o jogador fica parado.

Em cada ciclo do simulador ocorrem dois tipos de transição. Primeiro o simulador executa a ação de cada jogador, e segundo, a bola move-se de acordo com a direção e velocidade da bola. Para tratar as ações escolhidas por cada um dos agentes, o simulador

executa a ação de cada agente independentemente em ordem aleatória, assim o resultado das ações dos agentes que foram sorteados por último naquele ciclo depende do resultado das ações dos agentes que foram sorteados em primeiro naquele ciclo. Dessa forma, em um mesmo ciclo um jogador pode assumir a posição ocupada por outro jogador, se este mesmo jogador trocar de posição antes; ainda, um jogador pode recuperar a bola de um outro agente, mas no mesmo ciclo perder a bola. O monitor exibe apenas a configuração resultante no final do ciclo.

A ação *Stop* é a ação de permanecer na mesma posição atual. As ações da classe *Movimento* possuem probabilidade de transição `ProbPlayerMove` para se movimentar às posições adjacentes e dependem do ponto cardeal escolhido (Norte, Sul, Leste ou Oeste). Caso o jogador esteja com a bola, existe ainda a probabilidade `ProbBallWithPlayer` de a bola transitar para a nova posição. Portanto, um jogador pode deixar a bola para trás durante um movimento.

As ações da classe *Chute* ocorrem com probabilidade 1, mas a movimentação da bola é estocástica (ver mais abaixo). O jogador escolhe um ponto cardeal ou colateral (Nordeste, Sudeste, Noroeste ou Sudoeste), então uma velocidade `MaxBallVelocity` pré-configurada no simulador é imposta à bola.

A ação *Tackle*, por sua vez, é uma ação que permite aos jogadores imediatamente próximos ao possuidor da bola (i.e., nas 4 posições cardiais adjacentes) tentar tirá-la deste. A ação *Tackle* tem o mesmo funcionamento se a bola estiver adjacente ao jogador, mas em posse de ninguém, independente da velocidade da bola. Caso não esteja na proximidade da bola, essa ação é nula. Se um jogador está em posse da bola e ele sempre executa a ação *Stop*, a ação *Tackle* é a única forma de outro jogador tomar posse da bola. A ação *Tackle* é bem sucedida com probabilidade `ProbTackle`, quando a ação *Tackle* é bem sucedida a velocidade da bola é anulada.

Após executar as ações e respectivos movimento de todos os agentes, a movimentação da bola é executada. Se não houver nenhuma ocorrência negativa, a bola movimenta-se na direção da bola uma quantidade de posições dada pela velocidade da bola e a velocidade da bola é decrescida de uma unidade. As ocorrências negativas podem ser de várias maneiras: (i) a velocidade pode ser decrescida de uma unidade antes de ocorrer o movimento segundo a probabilidade `ProbBallDecay`; (ii) se a direção da bola for em diagonal (pontos colaterais), segundo uma probabilidade de desvio `ProbBallMoveDiagonal`, a bola pode andar em x ou/e y ; e (iii) caso a posição após a movimentação coincida com a posição de um jogador, a velocidade da bola é anulada com probabilidade de recuperar a bola `ProbTackle`, mesmo que esse jogador não tenha executado a ação *Tackle*.

2.1.3. Recompensa

Foi definida uma recompensa baseada na ocorrência de gol, expressa pela função:

$$\mathcal{R}(s, a) = \begin{cases} -1, & \text{se houve gol contra} \\ +1, & \text{se houve gol a favor} \\ 0, & \text{caso contrário.} \end{cases} \quad (1)$$

Note que a recompensa acumulada desse problema é a própria diferença de gols, critério para se vencer o jogo de futebol.

2.2. Controle da Partida

As duas variáveis de interesse modeladas são: (i) o contador de tempo, definido pela quantidade de iterações em que todos podem realizar ações; e (ii) o placar, obtido pela diferença da pontuação entre as equipes e que indica o vencedor. Uma função de espalhamento coloca todos os jogadores e a bola na posição inicial e define a posse de bola para o time oponente ao que marcou o último gol.

Como este simulador visa principalmente o uso científico e acadêmico, algumas opções foram implementadas para evitar situações de impasse. Primeiro, após um tempo *Horizon*, o simulador sorteia posições aleatórias para a bola e todos os jogadores; essa abordagem permite evitar impasses causados pelos controles implementados nos jogadores. Segundo, pode-se configurar a distribuição da bola e jogadores após um gol: aleatório ou padrão (jogadores aleatórios em seus respectivos lados do campo e bola no centro em posse de um jogador que sofreu o gol); aqui, a opção aleatória evita que estratégias de aprendizado ou controle baseado em regra fiquem dependente de uma situação padrão repetitiva.

2.3. Módulos

O simulador é dividido em 4 principais módulos: o núcleo, o motor gráfico, as estratégias de controle e o módulo principal (veja Figura 2).

O núcleo inclui: (i) o núcleo de simulação, (ii) o estado do problema, (iii) os agentes e (iv) as configurações do jogo e da simulação. O núcleo de simulação é a parte central na execução da simulação e implementa a dinâmica do simulador, i.e., é responsável pela requisição e execução da ação de cada jogador, bem como a atualização do estado mantendo as configurações definidas (e.g., tempo, placar).

O núcleo de simulação em cada iteração requisita a tomada de decisão de cada time, executa as ações e atualiza o estado do jogo. Cada time deve especificar uma ação para cada jogador do time e pode fazer isso de forma centralizada (considerando as ações em conjuntos) ou descentralizada (cada jogador escolhe sua própria ação); aqui, não existe um limite de tempo para tomada de decisão, de modo que o núcleo de simulação deve esperar cada um dos times tomar sua decisão.

O motor gráfico renderiza toda a visualização da simulação, que inclui os elementos estáticos do campo e os elementos dinâmicos da partida, como jogadores, bola, placar e tempo. Para acelerar experimentos, é possível controlar o tempo de exibição de cada ciclo do simulador, ou mesmo não exibir a parte gráfica.

O módulo de estratégias contem implementações concretas de estratégias de agentes, entre elas: aleatória, baseada em heurísticas e baseada em técnicas de aprendizado por reforço (descritas na Seção 3).

O módulo principal permite configurar e armazenar os problemas e invocar os elementos para simulação. Por exemplo, nesse módulo podem ser definidos os gráficos a serem plotados, as configurações das estratégias dos agentes, o comportamento do motor gráfico, salvar os agentes e variáveis simuladas, permitindo a pausa e continuidade de experimentos.

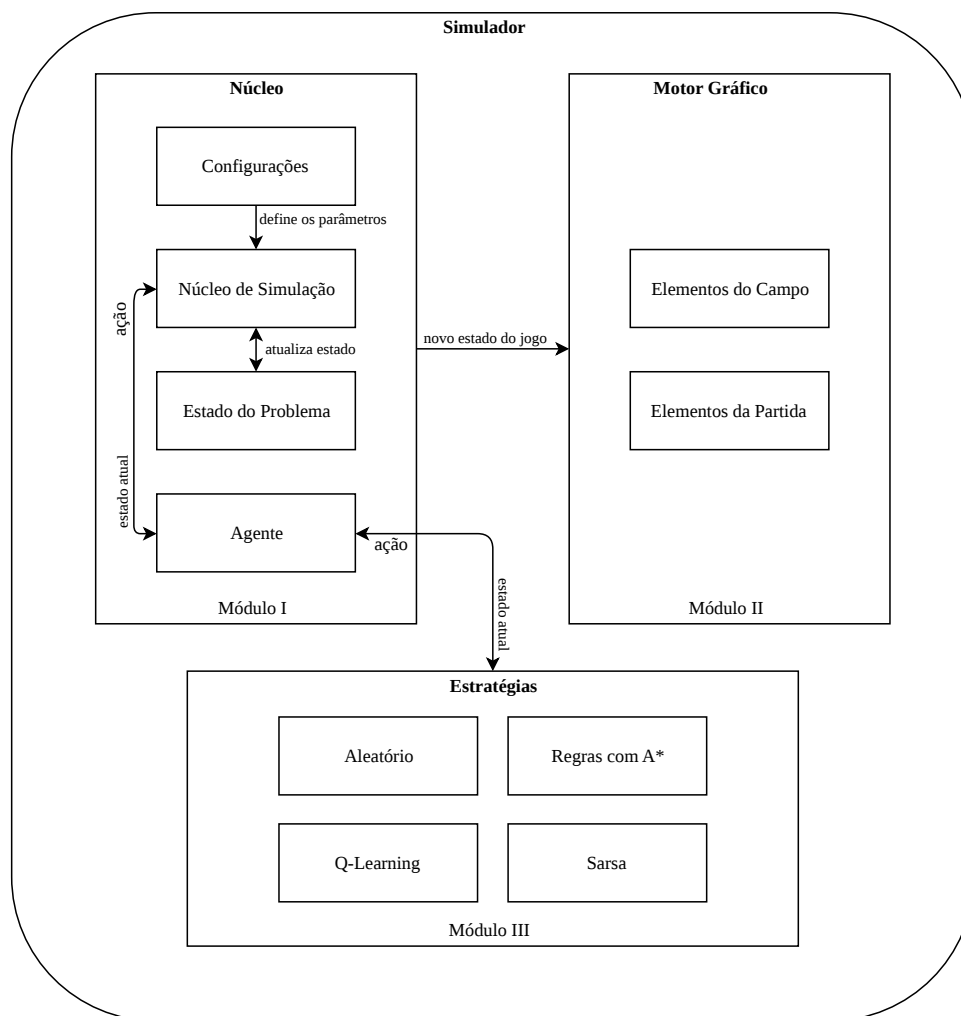


Figura 2. Organização do simulador D2DSS.

2.4. Configuração

Vários parâmetros podem ser utilizados para personalizar o ambiente de simulação e alguns já foram discutidos na Seção 2.1.2. Alguns desses parâmetros são fixados no simulador com valores padrão, mas podem ser facilmente alterados.

A Tabela 2 sumariza os parâmetros de configuração e apresenta o valor padrão configurado quando for o caso. As restrições que tais parâmetros devem respeitar são: (i) todos valores devem ser positivos, (ii) probabilidades devem ser menor ou igual a 1, (iii) N_x e N_y devem assumir um valor ímpar, e (iv) $N_x \times N_y > N_{playersA} + N_{playersB}$.

3. Estratégias de Controle

Assim como para o futebol humano, para simuladores de futebol faz-se necessário o contínuo desenvolvimento e aperfeiçoamento de estratégias que obtenham um melhor desempenho. As principais técnicas utilizadas para gerar estratégias na literatura são: aprendizado por reforço, heurísticas, algoritmos genéticos, sistemas fuzzy e redes neurais artificiais [Fraccaroli and da Silva 2009]. Neste trabalho, foram elaboradas algumas estratégias entre elas: aleatória, baseada em heurísticas e baseada em técnicas de aprendizado por reforço (Q-Learning e Sarsa).

Tabela 2. Parâmetros configuráveis no simulador.

Valor Padrão	Descrição
ProbBallWithPlayer 0.90	Probabilidade do jogador carregar a bola com ele quando está em posse da bola.
ProbPlayerMove 0.95	Probabilidade do jogador avançar na direção escolhida.
ProbTackle 0.50	Probabilidade do jogador recuperar a bola com a ação Tackle recuperar se o agente esta adjacente à bola.
ProbBallDecay 0.10	Probabilidade da velocidade da bola ser reduzida antes da bola se movimentar.
ProbBallMoveDiagonal 0.50	Probabilidade da bola andar em cada eixo x e y quando realiza chute na diagonal (probabilidade de desvio).
Nx	Altura do campo.
Ny	Largura do campo.
MaxBallVelocity $\left\lfloor \frac{\sqrt{3+8Nx}-\sqrt{3}}{2\sqrt{3}} \right\rfloor$	Velocidade impressa na bola após um chute.
SideGoalWidth $\left\lfloor \frac{Ny}{6} \right\rfloor$	Parâmetro para especificar a largura do gol, que é $2 \times SideGoalWidth + 1$. A altura do gol é sempre 1.
CellWidth 3	Tamanho de cada célula desenhada pelo motor gráfico.
Delay 0.05	O Atraso da atualização gráfica. Note que a esse atraso, soma-se o tempo gasto na decisão de cada time e pelo núcleo de simulação
Horizon $3(Nx + Ny)$	Horizonte de cada episódio.
NplayersA	Quantidade de jogadores no time A (lado esquerdo do simulador).
NplayersB	Quantidade de jogadores no time B (lado direito do simulador).

3.1. Aleatória

O agente escolhe uma ação de forma aleatória dentre as possíveis descritas na Tabela 1 e executa no estado atual.

3.2. Baseada em Heurística e Busca A*

O agente considera regras baseadas em heurísticas e Busca A* com uma quantidade máxima de nós explorados E . Em cada ciclo é escolhido qual jogador irá atrás da bola e as posições ideais dos outros jogadores, o controle do jogador é realizado de duas formas: (i) se uma condição de decisão é atendida, uma heurística simples é utilizada; e (ii) caso contrário, a Busca A* é utilizada para movimentar os jogadores para suas respectivas posições. Caso algum jogador esteja em posse da bola, com base na heurística ele pode tentar um passe, um chute ou melhorar sua posição usando uma das duas formas já citadas.

O jogador que vai atrás da bola é o jogador mais próximo da mesma. A posição ideal dos outros jogadores consiste em distribuir os jogadores igualmente espaçados ao longo do eixo x (profundidade do campo); enquanto que no eixo y (largura do campo) os jogadores que estão a frente da bola, simplesmente alinham-se à mesma no eixo y na espera de um passe, e os jogadores que estão atrás da bola alinham-se à mesma a partir do ponto do gol mais próximo da bola, para uma eventual defesa (veja Figura 3). A cada ciclo, é escolhido em ordem aleatória o jogador mais próximo de cada posição ideal e esse direciona-se para tal posição. Quando de posse da bola, o jogador segue algumas

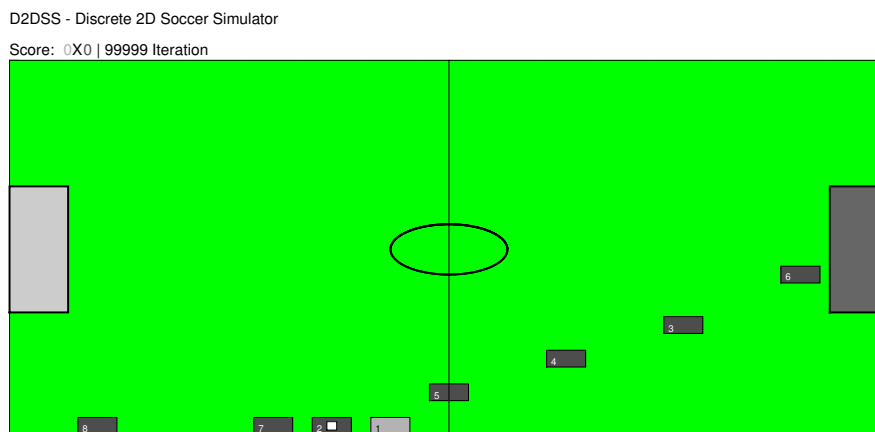


Figura 3. Exemplo de posições heurísticas da estratégia baseada em Regras e Busca A* em um campo 15×15 em um jogo de 7 jogadores contra 1 jogador.

heurísticas. Se existe a possibilidade de ao chutar a bola, a mesma entrar direto no gol (calculado por meio da velocidade máxima da bola), então o jogador realiza o chute na direção do gol. Se não é possível chutar ao gol, mas existe algum companheiro a frente dele no campo, simplesmente chuta a bola para frente, realizando um passe. Caso não se encontre em nenhuma das situações anteriores, o jogador tenta se aproximar de uma posição na qual ele pode chutar a bola ao gol, também escolhida heurísticamente.

A Busca A* é utilizada o mínimo possível e com profundidade limitada. Primeiro, considera-se posições alvos para cada jogador limitada a um quadrante de largura máxima L mesmo que a posição final não esteja dentro desse quadrante; a posição atual do jogador define um dos cantos do quadrante, dada as limitações do quadrante, o canto oposto é a posição mais próxima da posição final desejada. Esse quadrante permite verificar rapidamente se não existe adversário próximo ao agente, permitindo que a Busca A* seja evitada. Caso exista algum jogador no quadrante, então a Busca A* é chamada com profundidade L ; dessa forma, a Busca A* é utilizada apenas para evitar que jogadores bloqueiem um aos outros.

Dado que o simulador é estocástico e que existem outros agentes em campo, uma versão simplificada do simulador é utilizada para realizar a Busca A*. Primeiro, considera-se que todos os outros jogadores ficam parados; segundo, considera-se que ações de movimento são determinista. Dessa forma, a Busca A* pode ser utilizada para evitar ir em uma direção que está ocupada por outro agente, mas não pode evitar de passar próximo de outro agente e perder a bola.

No restante do texto chamaremos o agente que usa esta estratégia simplesmente de agente heurístico.

3.3. Agentes Baseados em Aprendizado por Reforço

Aprendizado por Reforço (*Reinforcement Learning* – RL) é um tipo de Aprendizado de Máquina muito utilizado por pesquisadores por não necessitar de ser supervisionado, i.e., não necessita de qualquer conhecimento *a priori* do ambiente que se quer explorar, nem mesmo exemplos de como agir, de forma que o aprendizado ocorra a partir da própria exploração incremental com o ambiente.[Sutton and Barto 1998]. Dois algoritmos foram implementados: O Q-Learning [Watkins and Dayan 1992] e Sarsa [Rummery and Niranjan 1994, Sutton and Barto 1998]. Existe ainda uma versão do Sarsa com aceleração de aprendizado por meio da seleção de atributos proposta por [da Silva and Freire 2016] para este simulador.

4. Experimentos

Nesta seção analisamos o desempenho do simulador, avaliando o seu comportamento com relação ao tempo de simulação e qualidade do agente heurístico. Os experimentos foram realizados com a versão 1.0 do Simulador D2DSS, no Matlab R2012a (7.14.0.739) 64 bits, em um Processador Intel Core i3-2100 CPU @3.10GHz, 6 GB de memória RAM. A interface gráfica foi desabilitada para o experimento.

4.1. Tempo de Simulação

Uma informação importante em um simulador multiagente, como é o caso do simulador D2DSS, diz respeito ao quanto o mesmo é escalável. A Figura 4 mostra curvas de tempo de execução para a quantidade de jogadores em cada time variando entre 1 e 10 com $L = 5$ e $E = 5$ em um campo 33 por 21. Pode-se ver que o tempo de execução aumenta linearmente com a quantidade de jogadores, tanto o tempo do simulador, quando o tempo para tomada de decisão. No entanto, o tempo de simulador tem uma inclinação maior.

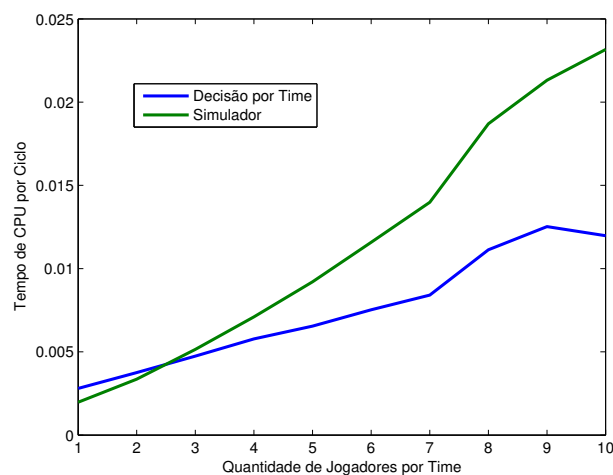


Figura 4. Tempo de CPU por Ciclo para várias quantidades de jogadores. O agente heurístico é configurado com $L = 5$, $E = 5$ e campo 33×21 .

Outra questão importante é como os parâmetros de limite L e quantidade de nós explorados E da estratégia heurística influenciam no tempo para tomada de decisão. A Figura 5 mostra curvas de tempo de execução para $L \in \{1, 5, 10\}$ variando o valor de E de 1 até 10. Enquanto, o tempo de execução é fortemente influenciado pelo limite L , a quantidade de nós explorados E pouco influencia no tempo de execução.

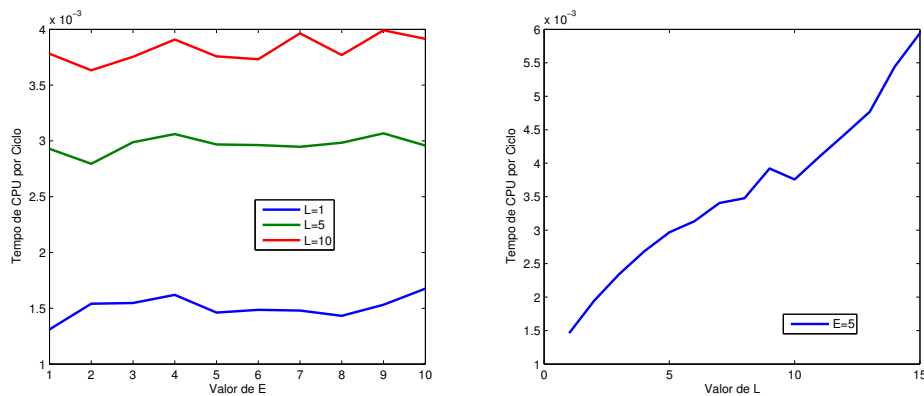


Figura 5. Tempo de CPU por Ciclo para tomada de decisão para vários valores de E e L para o agente heurístico. O simulador é configurado com 1 jogador em cada time e campo 33×21 .

4.2. Desempenho do Agente Heurístico

Embora vários agentes baseados em regras podem ser implementados no simulador D2DSS, o agente heurístico implementado pode ser utilizado como um adversário em testes de aprendizado ou planejamento. No entanto é importante saber como configurá-lo para que ele apresente o melhor desempenho com menor custo computacional. Na seção anterior já se viu que o parâmetro L influencia bastante no tempo de execução, enquanto o parâmetro E tem pouca influência.

Testes foram realizados com dois times de 1 agente heurístico. Os valores de E e L foram variados entre 1 e 5 para o time A , enquanto o time B utilizou $E = 5$ e $L = 5$. No entanto, nenhuma diferença relevante foi encontrada, a Figura 6 mostra histogramas e *boxplots* para vários valores de E com $L = 4$.

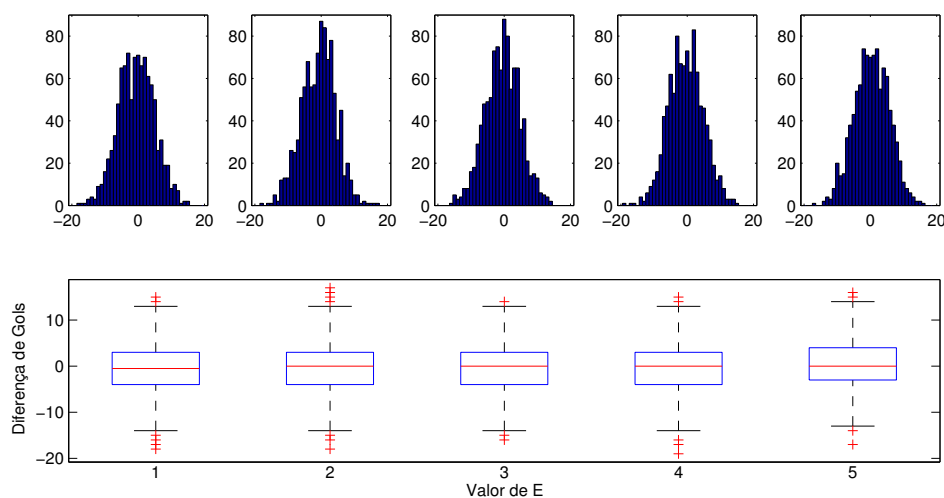


Figura 6. Histogramas e *Boxplots* para vários valores de E , com $L = 4$ para o agente heurístico. O simulador é configurado com 1 jogador em cada time e campo 33×21 .

Embora os parâmetros E e L não influenciam no desempenho do agente heurístico quando joga contra outro heurístico, isso não é necessariamente verdade quando joga contra outros agentes. Por exemplo, um agente de Aprendizado por Reforço pode aprender a tirar vantagem de limitações do agente heurístico. Experimentos de aprendizagem foram realizados para vários valores de E e L , enquanto no agente de Aprendizado por Reforço foi implementado o algoritmo Q-Learning com $\gamma = 0.99$, $\alpha = 0.1$ e $\epsilon = 0.1$; as médias de 20 execuções são exibidas na Figura 7. Nela pode-se ver que quando $L \geq 3$ o desempenho é similar, enquanto o agente baseado em RL pode tirar vantagem do agente heurístico se $L < 3$.

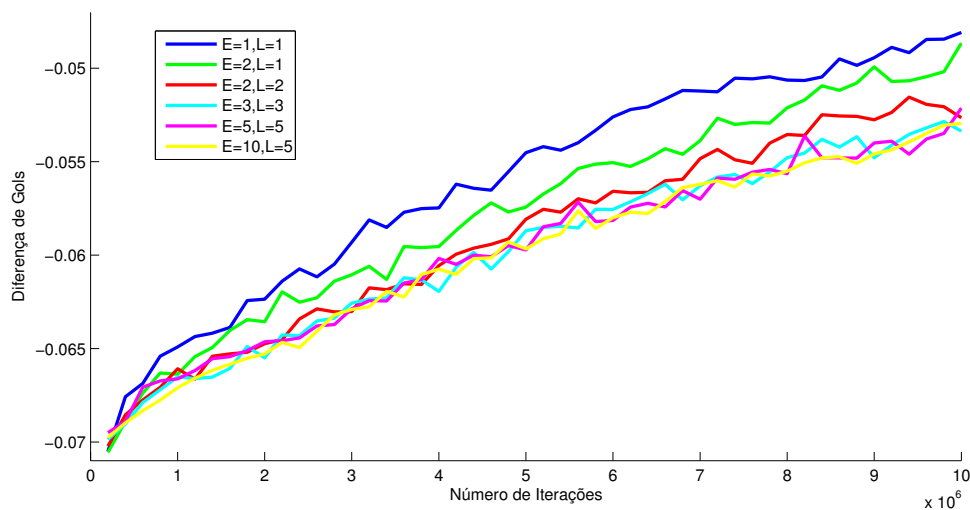


Figura 7. Curva de Aprendizagem do Agente Q-Learning para alguns valores de E e L para o agente heurístico. O simulador é configurado com 1 jogador em cada time e campo 11×7 .

5. Conclusão

Neste artigo, foi apresentado o simulador D2DSS. O objetivo é disponibilizar um simulador mais simples de ser utilizado, com pouca codificação e configurações para implementar um time. Também foi apresentado um agente heurístico, que pode ser utilizado como base de teste para algoritmos de aprendizagem ou planejamento.

Testes foram realizados com o simulador, mostrando que ele escala de forma linear com a quantidade de jogadores em campo. Testes também foram realizados com o agente heurístico, mostrando que em termos de tempo, o agente escala melhor que o simulador, e em termos de desempenho, pode escolher uma configuração ($L = 3$), que possui o mesmo desempenho quando utilizando valores maiores.

Finalmente, além de ser utilizado em pesquisas, o simulador D2DSS pode ser facilmente utilizado em sala de aula, uma vez que programar um primeiro agente demanda pouco tempo.

A escalabilidade do simulador pode ainda ser ampliada utilizando conceitos de processamento paralelo. Além disso, como trabalho futuro serão disponibilizadas as implementações de algoritmos clássicos de aprendizagem por reforço para fins didáticos.

Agradecimentos

Os autores agradecem à Capes (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela concessão da bolsa de mestrado para as atividades de pesquisa e à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo apoio financeiro (processo #2015/01587-0).

Referências

- [Akiyama et al. 2014] Akiyama, H., Dorer, K., and Lau, N. (2014). On the progress of soccer simulation leagues. In *Robot Soccer World Cup*, pages 599–610. Springer.
- [Chen et al. 2003] Chen, M., Dorer, K., Foroughi, E., Heintz, F., Huang, Z., Kapetanakis, S., Kostiadis, K., Kummeneje, J., Murray, J., Noda, I., Obst, O., Riley, P., Steffens, T., Wang, Y., and Yin, X. (2003). *Users Manual: RoboCup Soccer Server for Soccer Server Version 7.07 and Later*. The RoboCup Federation.
- [Costa and Leite 2012] Costa, S. and Leite, S. (2012). Implementação do simulador SIM para futebol de robôs. *Relatórios Técnicos do DCC/UFJF*.
- [da Silva and Freire 2016] da Silva, C. A. and Freire, V. (2016). Seleção de atributos no aprendizado por reforço avaliando o processo de aprendizagem. In *ENIAC 2016 (XIII Encontro Nacional de Inteligência Artificial e Computacional)*.
- [Fraccaroli and da Silva 2009] Fraccaroli, E. S. and da Silva, I. N. (2009). Abordagem genética para definição de ações em ambiente simulado de futebol de robôs. In *Anais do IX Simpósio Brasileiro de Automação Inteligente, Competições de Robôs Inteligentes*, pages 7–12.
- [Gabel and Riedmiller 2011] Gabel, T. and Riedmiller, M. (2011). *On Progress in RoboCup: The Simulation League Showcase*, pages 36–47. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Kitano et al. 1997] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., and Osawa, E. (1997). Robocup: The robot world cup initiative. In *Proceedings of the First International Conference on Autonomous Agents*, AGENTS '97, pages 340–347, New York, NY, USA. ACM.
- [Melo and Romero 2009] Melo, D. O. and Romero, R. A. F. (2009). O simulador 3D para futebol de robôs USPDS. *Relatórios Técnicos do ICMC/USP*.
- [Puterman 1994] Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition.
- [Rummery and Niranjan 1994] Rummery, G. and Niranjan, M. (1994). *On-line Q-learning Using Connectionist Systems*. CUED/F-INFENG/TR. University of Cambridge, Department of Engineering.
- [Sutton and Barto 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. A Bradford Book.
- [Watkins and Dayan 1992] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.