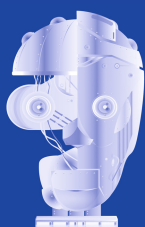


Jeng-Shyang Pan Marios M. Polycarpou
Michał Woźniak André C.P.L.F. de Carvalho
Héctor Quintián Emilio Corchado (Eds.)

LNAI 8073

Hybrid Artificial Intelligent Systems

8th International Conference, HAIS 2013
Salamanca, Spain, September 2013
Proceedings



HAIS
2013

 Springer

Lecture Notes in Artificial Intelligence 8073

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel

University of Alberta, Edmonton, Canada

Yuzuru Tanaka

Hokkaido University, Sapporo, Japan

Wolfgang Wahlster

DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann

DFKI and Saarland University, Saarbrücken, Germany

Jeng-Shyang Pan Marios M. Polycarpou
Michał Woźniak André C.P.L.F. de Carvalho
Héctor Quintián Emilio Corchado (Eds.)

Hybrid Artificial Intelligent Systems

8th International Conference, HAIS 2013
Salamanca, Spain, September 11-13, 2013
Proceedings

Volume Editors

Jeng-Shyang Pan

National Kaohsiung University of Applied Sciences, Taiwan R.O.C.

E-mail: jengshyangpan@gmail.com

Marios M. Polycarpou

University of Cyprus, Nicosia, Cyprus

E-mail: mpolycar@ucy.ac.cy

Michał Woźniak

Wrocław University of Technology, Poland

E-mail: michal.wozniak@pwr.wroc.pl

André C.P.L.F. de Carvalho

University of Sao Paulo at Sao Carlos, Brazil

E-mail: andre@icmc.usp.br

Héctor Quintián

University of Salamanca, Spain

E-mail: hector.quintian@usal.es

Emilio Corchado

University of Salamanca, Spain

E-mail: escorchado@usal.es

ISSN 0302-9743

e-ISSN 1611-3349

ISBN 978-3-642-40845-8

e-ISBN 978-3-642-40846-5

DOI 10.1007/978-3-642-40846-5

Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: Applied for

CR Subject Classification (1998): I.2, H.3, F.1, H.4, I.4, I.5

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume of *Lecture Notes in Artificial Intelligence* (LNAI) includes accepted papers presented at HAIS 2013 held in the beautiful and historic city of Salamanca, Spain, in September 2013.

The International Conference on Hybrid Artificial Intelligence Systems (HAIS) has become a unique, established, and broad interdisciplinary forum for researchers and practitioners who are involved in developing and applying symbolic and sub-symbolic techniques aimed at the construction of highly robust and reliable problem solving techniques, and bringing the most relevant achievements in this field.

Hybridization of intelligent techniques, coming from different computational intelligence areas, has become popular because of the growing awareness that such combinations frequently perform better than the individual techniques such as neurocomputing, fuzzy systems, rough sets, evolutionary algorithms, agents and multiagent systems, and alike.

Practical experience has indicated that hybrid intelligence techniques might be helpful for solving some of the challenging real-world problems. In a hybrid intelligence system, a synergistic combination of multiple techniques is used to build an efficient solution to deal with a particular problem. This is, thus, the setting of HAIS conference series, and its increasing success is the proof of the vitality of this exciting field.

HAIS 2013 received 218 technical submissions. After a rigorous peer-review process, the International Program Committee selected 68 papers that are published in this conference proceedings.

The selection of papers was extremely rigorous in order to maintain the high quality of the conference and we would like to thank the Program Committee for their hard work in the reviewing process. This process is very important to the creation of a conference of high standard and the HAIS conference would not exist without their help.

The large number of submissions is certainly not only testimony to the vitality and attractiveness of the field but an indicator of the interest in the HAIS conferences themselves.

HAIS 2013 enjoyed outstanding keynote speeches by distinguished guest speakers: Prof. Hojjat Adeli - Ohio State University (USA), Prof. Hujun Yin - University of Manchester (UK), and Prof. Manuel Graña - University of Pais Vasco (Spain).

HAIS 2013 teamed up with the *International Journal of Neural Systems* (WORLD SCIENTIFIC), *Integrated Computer-Aided Engineering* (IOS PRESS), *Neurocomputing* (ELSEVIER), and the *Applied Soft Computing* (ELSEVIER) journals for a set of special issues and fast track including selected papers from HAIS 2013.

Particular thanks also go to the conference main Sponsors, IEEE-Sección España, IEEE Systems, Man and Cybernetics –Capítulo Español, AEPIA, Ayuntamiento de Salamanca, University of Salamanca, World Federation of Soft Computing, MIR Labs, IT4Innovation Centre of Excellence, The International Federation for Computational Logic, Ministerio de Economía y Competitividad (TIN 2010-21272-C02-01), Junta de Castilla y León (SA405A12-2), INMOTIA, REPLENTIA, and HIDROGENA, who jointly contributed in an active and constructive manner to the success of this initiative.

We would like to thank Alfred Hofmann and Anna Kramer from Springer for their help and collaboration during this demanding publication project.

September 2013

Jeng-Shyang Pan
Marios Polycarpou
Michał Woźniak
André C.P.L.F. de Carvalho
Héctor Quintián
Emilio Corchado

Organization

Honorary Chairs

Alfonso Fernández Mañueco	Mayor of Salamanca
Costas Stasopoulos	Director-Elect. IEEE Region 8
Antonio Bahamonde	President of the Spanish Association for Artificial Intelligence (AEPIA)
Pilar Molina	IEEE Spanish Section President

General Chair

Emilio Corchado	University of Salamanca, Spain
-----------------	--------------------------------

International Advisory Committee

Ajith Abraham	Machine Intelligence Research Labs, Europe
Antonio Bahamonde	President of the Spanish Association for Artificial Intelligence, AEPIA
Andre de Carvalho	University of São Paulo, Brazil
Sung-Bae Cho	Yonsei University, Korea
Juan M. Corchado	University of Salamanca, Spain
José R. Dorronsoro	Autonomous University of Madrid, Spain
Michael Gabbay	Kings College London, UK
Ali A. Ghorbani	UNB, Canada
Mark A. Girolami	University College London, UK
Manuel Graña	University of País Vasco, Spain
Petro Gopych	Universal Power Systems USA-Ukraine LLC, Ukraine
Jon G. Hall	The Open University, UK
Francisco Herrera	University of Granada, Spain
César Hervás-Martínez	University of Córdoba, Spain
Tom Heskes	Radboud University Nijmegen, The Netherlands
Dusan Husek	Academy of Sciences of the Czech Republic, Czech Republic
Lakhmi Jain	University of South Australia, Australia
Samuel Kaski	Helsinki University of Technology, Finland
Daniel A. Keim	University of Konstanz, Germany
Isidro Laso	D.G. Information Society and Media, European Commission
Marios Polycarpou	University of Cyprus, Cyprus

VIII Organization

Witold Pedrycz	University of Alberta, Canada
Václav Snášel	VSB-Technical University of Ostrava, Czech Republic
Xin Yao	University of Birmingham, UK
Hujun Yin	University of Manchester, UK
Michał Woźniak	Wroclaw University of Technology, Poland
Aditya Ghose	University of Wollongong, Australia
Ashraf Saad	Armstrong Atlantic State University, USA
Fanny Klett	German Workforce Advanced Distributed Learning Partnership Laboratory, Germany
Paulo Novais	Universidade do Minho, Portugal

Industrial Advisory Committee

Rajkumar Roy	The EPSRC Centre for Innovative Manufacturing in Through-life Engineering Services (UK)
Amy Neustein	Linguistic Technology Systems, USA
Francisco Martinez	INMOTIA

Program Committee

Emilio Corchado	University of Salamanca, Spain (Co-chair)
Jeng-Shyang Pan	National Kaohsiung University of Applied Sciences, Taiwan (Co-chair)
Marios Polycarpou	University of Cyprus, Cyprus, Cyprus (Co-chair)
Michał Woźniak	Wroclaw University of Technology, Poland (Co-chair)
André C.P.L.F. de Carvalho	University of São Paulo, Brazil (Co-chair)
Abdel-Badeeh Salem	Ain Shams University, Egypt
Aboul Ella Hassanien	Cairo University, Egypt
Adolfo R. De Soto	University of León, Spain
Ajith Abraham	Machine Intelligence Research Labs (MIR Labs), Europe
Alberto Fernandez Gil	University Rey Juan Carlos, Spain
Alicia Troncoso	University Pablo de Olavide, Spain
Álvaro Herrero	University of Burgos, Spain
Amelia Zafra Gómez	University of Córdoba, Spain
Ana Madureira	Polytechnic University of Porto, Portugal
Ana M. Bernardos	Polytechnic University of Madrid, Spain
Anca Andreica	Babes-Bolyai University, Romania
Andreea Vescan	Babes-Bolyai University, Romania
Andrés Ortíz	University of Malaga, Spain

Ángel Arroyo	University of Burgos, Spain
Angelos Amanatiadis	Democritus University of Thrace, Greece
Arkadiusz Kowalski	Wroclaw University of Technology, Poland
Arturo De La Escalera	University Carlos III of Madrid, Spain
Bogdan Trawinski	Wroclaw University of Technology, Poland
Borja Fernandez-Gauna	University of Basque Country / EHU, Spain
Bożena Skołod	Silesian University of Technology, Poland
Bruno Baruque	University of Burgos, Spain
Camelia Chira	Babes-Bolyai University, Romania
Camelia Pinte	George Coşbuc N College, Cluj-Napoca, Romania
Carlos Pereira	University of Coimbra, Portugal
Carlos Laorden	University of Deusto, Spain
Carlos Carrascosa	Polytechnic University of Valencia, Spain
Carlos Laorden	University of Deusto, Spain
Carmen Vidaurre	Berlin Institute of Technology, Germany
Cerasela Crisan	University of Bacau, Romania
Cesare Alippi	Politecnico di Milano, Italy
Cezary Grabowik	Silesian University of Technology, Poland
Constantin Zopounidis	University of Crete, Greece
Cristina Rubio-Escudero	University of Seville, Spain
Cristobal J. Carmona	University of Jaen, Spain
Damian Krenczyk	Silesian University of Technology, Poland
Dario Landa-Silva	University of Nottingham, UK
Darya Chyzhyk	University of Basque Country / EHU, Spain
David Iclanzan	Sapientia Hungarian University of Transylvania, Romania
Diego Pablo Ruiz	University of Granada, Spain
Donald Davendra	VSB - Technical University of Ostrava, Czech Republic
Dragan Simic	University of Novi Sad, Serbia
Dragos Horvath	University of Strassbourg, France
Eiji Uchino	Yamaguchi University, Japan
Estefania Argente	Polytechnic University of Valencia, Spain
Eva Volna	University of Ostrava, Czech Republic
Fab�ricio Olivetti De Fran�a	UNICAMP, Brazil
Federico Divina	University Pablo de Olavide, Spain
Fermin Segovia	University of Granada, Spain
Fernando De La Prieta	University of Salamanca, Spain
Fidel Aznar	University of Alicante, Spain
Florentino Fdez-Riverola	University of Vigo, Spain
Francisco Mart�nez-�lvarez	University Pablo de Olavide, Spain
Francisco Bellas	University of Coru�a, Spain
Francisco Cuevas	CIO, M�xico

Frank Klawonn	Ostfalia University of Applied Sciences, Germany
George Papakostas	Democritus University of Thrace, Greece
Georgios Dounias	University of the Aegean, Greece
Giancarlo Mauri	University of Milano-Bicocca, Italy
Giorgio Fumera	University of Cagliari, Italy
Guiomar Corral Torruella	Enginyeria i Arquitectura La Salle, Spain
Guoyin Wang	Chongqing University of Posts and Telecommunications, China
Héctor Quintián	University of Salamanca, Spain
Henrietta Toman	University of Debrecen, Hungary
Ignacio Turias	University of Cadiz, Spain
Igor Santos	University of Deusto, Spain
Ines Galvan	University Carlos III of Madrid, Spain
Ingo R. Keck	University of Regensburg, Germany
Ioannis Hatzilygeroudis	University of Patras, Greece
Irene Diaz	University of Oviedo, Spain
Isabel Barbancho	University of Málaga, Spain
Isabel Nepomuceno	University of Seville, Spain
Isabel Barbancho	University of Málaga, Spain
Jacinto Mata	University of Huelva, Spain
Jan Platos	VSB - Technical University of Ostrava, Czech Republic
Jaume Bacardit	University of Nottingham, UK
Javier Sedano	ITCL, Spain
Javier Bajo	Polytechnic University of Madrid, Spain
Javier De Lope	Polytechnic University of Madrid, Spain
Jeng-Shyang Pan Pan	National Kaohsiung University of Applied Sciences, China
Jesús Alcalá-Fdez	University of Granada, Spain
Joaquín Derrac	University of Granada, Spain
José Dorronsoro	Autonomous University of Madrid, Spain
José García-Rodríguez	University of Alicante, Spain
José C. Riquelme Santos	University of Seville, Spain
José Luis Calvo Rolle	University of Coruña, Spain
José Luis Verdegay	University of Granada, Spain
José M. Molina	University Carlos III of Madrid, Spain
José Manuel Lopez-Guede	University of Basque Country / EHU, Spain
José María Armingol	University Carlos III of Madrid, Spain
José Ramón Villar	University of Oviedo, Spain
José-Ramón Cano De Amo	University of Jaen, Spain
José Ranilla	University of Oviedo, Spain
Juan Pavón	Complutense University of Madrid
Juan Álvaro Muñoz Naranjo	University of Almería, Spain
Juan Humberto Sossa Azuela	National Polytechnic Institute, Mexico

Julián Luengo	University of Granada, Spain
Julio Ponce	Autonomous University of Aguascalientes, Mexico
Krzysztof Kalinowski	Silesian University of Technology, Poland
Lars Graening	Honda Research Institute Europe GmbH, Germany
Laura García-Hernández	University of Córdoba, Spain
Lauro Snidaro	University of Udine, Italy
Lenka Lhotska	Czech Technical University in Prague, Czech Republic
Leocadio G. Casado	University of Almeria, Spain
Lourdes Saíz Bárcena	University of Burgos, Spain
Manuel Graña	University of Basque Country / EHU, Spain
Marcilio De Souto	LIFO/University of Orleans, France
Marcin Zmysłony	Wroclaw University of Technology, Poland
María Guijarro	Complutense University of Madrid, Spain
María Martínez-Ballesteros	University of Seville, Spain
María José Del Jesus	University of Jaén, Spain
María R Sierra	University of Oviedo, Spain
Mario Koeppen	Kyushu Institute of Technology, Japan
Martí Navarro	Polytechnic University of Valencia, Spain
Martin Macas	Czech Technical University in Prague, Czech Republic
Matjaz Gams	Jozef Stefan Institute, Slovenia
Miguel Ángel Patricio	University Carlos III of Madrid, Spain
Miguel Ángel Veganzones	GIPSA-lab, Grenoble INP, France
Milos Kudelka	VSB - Technical University of Ostrava, Czech Republic
Miroslav Bursa	Czech Technical University in Prague, Czech Republic
Nicola Di Mauro	Università di Bari, Italy
Nima Hatami	University of California, USA
Noelia Sanchez-Maróño	University of Coruña, Spain
Oscar Fontenla-Romero	University of Coruña, Spain
Ozgur Koray Sahingoz	Turkish Air Force Academy, Turkey
Paula M. Castro Castro	University of Coruña, Spain
Paulo Novais	University of Minho, Portugal
Pavel Kromer	VSB - Technical University of Ostrava, Czech Republic
Pavel Brandstetter	VSB - Technical University of Ostrava, Czech Republic
Peter Rockett	University of Sheffield, UK
Peter Sussner	UNICAMP, Brazil
Petrica Claudiu Pop	North University of Baia Mare, Romania
Przemyslaw Kazienko	Wroclaw University of Technology, Poland

Rafael Corchuelo	University of Seville, Spain
Ramón Rizo	University of Alicante, Spain
Ramón Moreno	University of Basque Country / EHU, Spain
Ricardo Del Olmo	University of Burgos, Spain
Robert Burduk	Wroclaw University of Technology, Poland
Rodolfo Zunino	University of Genova, Italy
Roman Senkerik	TBU in Zlin, Czech Republic
Rubén Fuentes-Fernández	Complutense University of Madrid, Spain
Sean Holden	University of Cambridge, UK
Sebastián Ventura	University of Córdoba, Spain
Sooyoung Lee	KAIST, South Korea
Stella Heras	Polytechnic University of Valencia, Spain
Sung-Bae Cho	Yonsei University, South Korea
Talbi El-Ghazali	University of Lille, France
Theodore Pachidis	Kavala Institute of Technology, Greece
Tomasz Kajdanowicz	Wroclaw University of Technology, Poland
Urko Zurutuza	Mondragon University, Spain
Urszula Stanczyk	Silesian University of Technology, Poland
Václav Snášel	VSB - Technical University of Ostrava, Czech Republic
Vicente Martin-Ayuso	Polytechnic University of Madrid, Spain
Waldemar Malopolski	Tadeusz Kościuszko Cracow University of Technology, Poland
Wei-Chiang Hong	Oriental Institute of Technology, China
Wiesław Chmielnicki	Jagiellonian University, Poland
Yannis Marinakis	Technical University of Crete, Greece
Ying Tan	Peking University, China
Yusuke Nojima	Osaka Prefecture University, Japan
Zuzana Oplatkova	Tomas Bata University in Zlin, Czech Republic

Organizing Committee

Emilio Corchado	University of Salamanca, Spain
Álvaro Herrero	University of Burgos, Spain
Bruno Baruque	University of Burgos, Spain
Héctor Quintián	University of Salamanca, Spain
Roberto Vega	University of Salamanca, Spain
José Luis Calvo	University of Coruña, Spain
Ángel Arroyo	University of Burgos, Spain
Laura García-Hernández	University of Cordoba, Spain

Table of Contents

Agents and Multi Agents Systems

An Agent Based Implementation of Proactive S-Metaheuristics	1
<i>Mailyn Moreno, Alejandro Rosete, and Juan Pavón</i>	
An Ontological and Agent-Oriented Modeling Approach for the Specification of Intelligent Ambient Assisted Living Systems for Parkinson Patients	11
<i>Iván García-Magariño and Jorge J. Gómez-Sanz</i>	
Integration of Self-organization and Cooperation Mechanisms to Enhance Service Discovery	21
<i>Elena del Val, Miguel Rebollo, and Vicente Botti</i>	
Agent Participation in Context-Aware Workflows	31
<i>José M. Fernández-de-Alba, Rubén Fuentes-Fernández, and Juan Pavón</i>	
PHAT: Physical Human Activity Tester	41
<i>Pablo Campillo-Sanchez, Jorge J. Gómez-Sanz, and Juan A. Botía</i>	

HAIS Applications

Support Vector Forecasting of Solar Radiation Values	51
<i>Yvonne Gala, Ángela Fernández, Julia Díaz, and José R. Dorronsoro</i>	
A Hybrid Fuzzy Approach to Facility Location Decision-Making	61
<i>Dragan Simić, Vasa Svirčević, and Svetlana Simić</i>	
Clinical Careflows Aided by Uncertainty Representation Models	71
<i>Tiago Oliveira, João Neves, Ernesto Barbosa, and Paulo Novais</i>	
A Hybrid Approach for the Verification of Integrity Constraints in Clinical Practice Guidelines	81
<i>Marco Iannaccone, Massimo Esposito, and Giuseppe De Pietro</i>	
Hippocampus Localization Guided by Coherent Point Drift Registration Using Assembled Point Set	92
<i>Anusha Achuthan, Mandava Rajeswari, and Win Mar @ Salmah Jalaluddin</i>	

Classification and Cluster Analysis

Network Anomaly Classification by Support Vector Classifiers Ensemble and Non-linear Projection Techniques	103
<i>Eduardo de la Hoz, Andrés Ortiz, Julio Ortega, and Emiro de la Hoz</i>	
Classification Method for Differential Diagnosis Based on the Course of Episode of Care	112
<i>Adrian Popiel, Tomasz Kajdanowicz, Przemyslaw Kazienko, Jean Karl Soler, Derek Corrigan, Vasa Curcin, Roxana Danger Mercaderes, and Brendan Delaney</i>	
Movie Recommendation Framework Using Associative Classification and a Domain Ontology	122
<i>María N. Moreno, Saddys Segrera, Vivian F. López, María Dolores Muñoz, and Angel Luis Sánchez</i>	
Construction of Sequential Classifier Based on Broken Stick Model	132
<i>Robert Burduk and Pawel Trajdos</i>	
Model and Feature Selection in Hidden Conditional Random Fields with Group Regularization	140
<i>Rodrigo Cilla, Miguel A. Patricio, Antonio Berlanga, and José M. Molina</i>	

Data Mining and Knowledge Discovery

A First Approach to Deal with Imbalance in Multi-label Datasets	150
<i>Francisco Charte, Antonio Rivera, María José del Jesus, and Francisco Herrera</i>	
Simulating a Collective Intelligence Approach to Student Team Formation	161
<i>Juan M. Alberola, Elena del Val, Victor Sanchez-Anguix, and Vicente Julian</i>	
A Counting-Based Heuristic for ILP-Based Concept Discovery Systems	171
<i>Alev Mutlu, Pınar Karagoz, and Yusuf Kavurucu</i>	
Extracting Sequential Patterns Based on User Defined Criteria	181
<i>Oznur Kirmemis Alkan and Pınar Karagoz</i>	
Sequence Alignment Adaptation for Process Diagnostics and Delta Analysis	191
<i>Eren Esgin and Pınar Karagoz</i>	

Qualitative Reasoning on Complex Systems from Observations	202
<i>Gonzalo A. Aranda-Corral, Joaquín Borrego-Díaz, and Juan Galán-Páez</i>	
Reference Data Sets for Spam Detection: Creation, Analysis, Propagation	212
<i>Marcin Luckner and Robert Filasiak</i>	
Monitoring Mental Fatigue through the Analysis of Keyboard and Mouse Interaction Patterns	222
<i>André Pimenta, Davide Carneiro, Paulo Novais, and José Neves</i>	
On Mining Sensitive Rules to Identify Privacy Threats	232
<i>Irene Díaz, Luis J. Rodríguez-Muniz, and Luigi Troiano</i>	
An Evidential and Context-Aware Recommendation Strategy to Enhance Interactions with Smart Spaces	242
<i>Josué Iglesias, Ana M. Bernardos, and José R. Casar</i>	
Information Fusion for Context Awareness in Intelligent Environments	252
<i>Fábio Silva, Cesar Analide, and Paulo Novais</i>	
Simply-Integrated Method of Judgments of Expert Knowledge Collected in Databases for Objective Computer-Aided Engineering Systems	262
<i>Piotr Michalski, Mariusz Piotr Hetmańczyk, and Jerzy Świder</i>	
A Hybrid Inference Approach for Building Fuzzy DSSs Based on Clinical Guidelines	269
<i>Aniello Minutolo, Massimo Esposito, and Giuseppe De Pietro</i>	
Hybrid Visualization for Deep Insight into Knowledge Retention in Firms	280
<i>Lourdes Sáiz, Miguel A. Manzanedo, Arturo Pérez, Álvaro Herrero, and Emilio Corchado</i>	
Video and Image Analysis	
Fall Detection Using Kinect Sensor and Fall Energy Image	294
<i>Bogdan Kwolek and Michal Kepski</i>	
Modified Dendrite Morphological Neural Network Applied to 3D Object Recognition on RGB-D Data	304
<i>Humberto Sossa and Elizabeth Guevara</i>	
Diversity Measures for Majority Voting in the Spatial Domain	314
<i>Andras Hajdu, Lajos Hajdu, Laszlo Kovacs, and Henrietta Toman</i>	

How Do You Help a Robot to Find a Place? A Supervised Learning Paradigm to Semantically Infer about Places 324
Ioannis Kostavelis, Angelos Amanatiadis, and Antonios Gasteratos

Study of the Pre-processing Impact in a Facial Recognition System 334
Guillermo Calvo, Bruno Baruque, and Emilio Corchado

Bio-inspired Models and Evolutionary Computation

Using ABC Algorithm with Shrinkage Estimator to Identify Biomarkers of Ovarian Cancer from Mass Spectrometry Analysis 345
Syarifah Adilah Mohamed Yusoff, Rosni Abdullah, and Ibrahim Venkat

Metaoptimization of Differential Evolution by Using Productions of Low-Number of Cycles: The Fitting of Rotation Curves of Spiral Galaxies as Case Study 356
Miguel Cárdenas-Montes, Miguel Á. Vega-Rodríguez, and Mercedes Mollá

The Artificial Bee Colony Algorithm Applied to a Self-adaptive Grid Resources Selection Model 366
María Botón-Fernández, Miguel Á. Vega-Rodríguez, and Francisco Prieto Castrillo

A Hybrid Algorithm Combining an Evolutionary Algorithm and a Simulated Annealing Algorithm to Solve a Collaborative Learning Team Building Problem 376
Virginia Yannibelli and Analía Amandi

Addressing Constrained Sampling Optimization Problems Using Evolutionary Algorithms 390
Pilar Caamaño, Gervasio Varela, and Richard J. Duro

Genetic Algorithm-Based Allocation and Scheduling for Voltage and Frequency Scalable X MOS Chips 401
Zorana Banković and Pedro López-García

Second Order Swarm Intelligence 411
Vitorino Ramos, David M.S. Rodrigues, and Jorge Louçã

Learning Algorithms

Hybrid Approach Using Rough Sets and Fuzzy Logic to Pattern Recognition Task 421
Andrzej Zolnierak and Marcin Majak

MLG: Enhancing Multi-label Classification with Modularity-Based Label Grouping	431
<i>Piotr Szymański and Tomasz Kajdanowicz</i>	
Intelligent System for Channel Prediction in the MIMO-OFDM Wireless Communications Using a Multidimensional Recurrent LS-SVM	441
<i>Jerzy Martyna</i>	
Template-Based Synthesis of Plan Execution Monitors	451
<i>Thomas Reinbacher and César Guzmán-Alvarez</i>	
Distributed Privacy-Preserving Minimal Distance Classification	462
<i>Bartosz Krawczyk and Michał Woźniak</i>	

Systems, MAN, and CYBERNETICS

Borderline Kernel Based Over-Sampling	472
<i>María Pérez-Ortiz, Pedro Antonio Gutiérrez, and Cesar Hervás-Martínez</i>	
Discrimination of Resting-State fMRI for Schizophrenia Patients with Lattice Computing Based Features	482
<i>Darya Chyzyk and Manuel Graña</i>	
Enhancing Active Learning Computed Tomography Image Segmentation with Domain Knowledge	491
<i>Borja Ayerdi, Josu Maiora, and Manuel Graña</i>	
Evolutionary Ordinal Extreme Learning Machine	500
<i>Javier Sánchez-Monedero, Pedro Antonio Gutiérrez, and Cesar Hervás-Martínez</i>	
Arm Orthosis/Prosthesis Control Based on Surface EMG Signal Extraction	510
<i>Aaron Suberbiola, Ekaitz Zulueta, Jose Manuel Lopez-Guede, Ismael Etxeberria-Agiriano, and Bren Van Caesbroeck</i>	
Application Possibilities of Hardware Implemented Hybrid Neural Networks to Support Independent Life of Elderly People	520
<i>Stefan Oniga and Petrica Pop-Sitar</i>	
Multi-agent Reactive Planning for Solving Plan Failures	530
<i>César Guzmán-Alvarez, Pablo Castejon, Eva Onaindia, and Jeremy Frank</i>	
A Discussion on Trust Requirements for a Social Network of Eahoukers	540
<i>Manuel Graña, J. David Nuñez-Gonzalez, and Bruno Apolloni</i>	

Hybrid Intelligent Systems for Data Mining and Applications

Querying on Fuzzy Surfaces with Vague Queries	548
<i>Jan Caha and Jiří Dvorský</i>	
Best Fuzzy Partitions to Build Interpretable DSSs for Classification in Medicine	558
<i>Marco Pota, Massimo Esposito, and Giuseppe De Pietro</i>	
An Experimental Case of Study on the Behavior of Multiple Classifier Systems with Class Noise Datasets	568
<i>José A. Sáez, Mikel Galar, Julián Luengo, and Francisco Herrera</i>	
A Sensitivity Analysis for Quality Measures of Quantitative Association Rules	578
<i>María Martínez-Ballesteros, Francisco Martínez-Álvarez, Alicia Troncoso, and José C. Riquelme</i>	
Building a Robust Extreme Learning Machine for Classification in the Presence of Outliers	588
<i>Ana Luíza B.P. Barros and Guilherme A. Barreto</i>	
Handling Inconsistencies in the Revision of Probability Distributions . . .	598
<i>Fabian Schmidt, Jan Wendler, Jörg Gebhardt, and Rudolf Kruse</i>	
Creating Knowledge Base from Automatically Extracted Information . . .	608
<i>Beata Nachyta</i>	
A HMM-Based Location Prediction Framework with Location Recognizer Combining k-Nearest Neighbor and Multiple Decision Trees	618
<i>Yong-Joong Kim and Sung-Bae Cho</i>	
Noisy Data Set Identification	629
<i>Luís Paulo F. García, André C.P.L.F. de Carvalho, and Ana C. Lorena</i>	
Density-Based Clustering in Cloud-Oriented Collaborative Multi-Agent Systems	639
<i>Jelena Fiosina and Maksims Fiosins</i>	

Metaheuristics for Combinatorial Optimization and Modelling Complex Systems

A Hybrid Genetic Algorithm with Variable Neighborhood Search Approach to the Number Partitioning Problem	649
<i>Levente Fuksz and Petrica C. Pop</i>	

Human Activity Recognition and Feature Selection for Stroke Early Diagnosis	659
<i>José Ramón Villar, Silvia González, Javier Sedano, Camelia Chira, and José M. Trejo</i>	
Using a Hybrid Cellular Automata Topology and Neighborhood in Rule Discovery	669
<i>Anca Andreica and Camelia Chira</i>	
An Extension of the FURIA Classification Algorithm to Low Quality Data	679
<i>Ana Maria Palacios, Luciano Sanchez, and Ines Couso</i>	
Author Index	689

An Agent Based Implementation of Proactive S-Metaheuristics

Mailyn Moreno¹, Alejandro Rosete¹, and Juan Pavón²

¹ Facultad de Ingeniería Informática, Instituto Superior Politécnico “José Antonio Echeverría”
(CUJAE), La Habana, Cuba

{my, rosete}@ceis.cujae.edu.cu

² Dep. Ingeniería del Software e Inteligencia Artificial,
Universidad Complutense de Madrid, Spain

jpavon@fdi.ucm.es

Abstract. This paper presents the use of a multi-agent system for the development of proactive S-Metaheuristics (i.e. single-solution based metaheuristics) derived from Record-to-Record Travel (RRT) and Local Search. The basic idea is to implement metaheuristics as agents that operate in the environment of the optimization process with the goal of avoiding stagnation in local optima by adjusting their parameters and neighborhood. Environmental information about previous solutions is used to determine the best operators and parameters. The proactive adjustment of the neighborhood is based on the identification of the best operators using Fitness Distance Correlation (FDC). The proactive adjustment of the parameters is focused on guarantying a minimal level of acceptance of new solutions. Besides, a simple form of combination of both proactive behaviors is introduced. The system has been validated through experimentation with 28 functions on binary strings.

Keywords: Metaheuristics, Agents, Proactivity, Local Search, RRT, FDC.

1 Introduction

Metaheuristics are popular optimization methods due to their ability to find good solutions (not necessarily optimal) to complex optimization problems in different domains [1]. Local Search (LS) is an important root in the genealogy of metaheuristics [1] that iteratively improves a solution according to the criteria to be optimized. The principal problem of LS is the convergence to local (not global) optima. The existence of local optima is the consequence of two aspects: operators (neighborhood) and acceptance criterion. Many S-Metaheuristics (single-solution based metaheuristics) [1] have been designed to overcome this problem by relaxing the acceptance criterion (some worse solutions are accepted) or by modifying the neighborhood. In all these metaheuristics, several parameters need to be adjusted to get good results. Besides, according to the No Free Lunch (NFL) theorem, it is impossible to demonstrate that one metaheuristic outperforms all the others in all possible problems [2]. Several predictive measures of problem difficulty (e.g. Fitness Distance Correlation

(FDC) [3]) have been proposed to learn which characteristics of a problem make it difficult for certain metaheuristic.

This paper is focused in developing proactive S-Metaheuristics that behave proactively (i.e., adjusting themselves the parameters and neighborhood), according to the goals of the optimizer. We use the i^* language [5] to model S-Metaheuristics as agents that act in an environment (optimization process) with the goal of achieving a global optimum, while avoiding local optima. This facilitates the identification of goals, and plans to incorporate proactivity. The use of a multi-agent system provides flexibility in the solution as agents can negotiate among them and adjust parameters. The system evolves through a series of iterations by considering previous solutions to detect the best parameter settings and neighborhood structure. Section 2 explains the main concepts of agents and metaheuristics that are relevant to this paper. Section 3 presents the analysis and design of the system model with the i^* methodology. This model applies proactive adjustment of parameters and neighborhoods, based on the information gathered from the environment. Section 4 presents an experimental validation of the proactive metaheuristics in 28 functions on 100-bits strings. Section 5 presents the conclusions and discusses possible extensions to this approach.

2 S-Metaheuristics and Agents

2.1 S-Metaheuristics: Parameters, Neighborhoods, and Measures

S-Metaheuristics are single-solution based metaheuristics [1], which use the current (single) solution as a reference, in order to generate new solutions by consecutive applications of the operators. All S-Metaheuristics keep the best solution found during the course of the optimization process. The performance of metaheuristics depends highly on the balance between two factors: exploration and exploitation [1].

Random Search (RS) is located in one extreme of exploration, because every new solution is generated without any considerations of the previously generated solutions. Local Search (LS) is in the other extreme, because it generates new solutions as modifications of the best previous solutions. A new solution is only accepted as a reference to generate new ones if it is better than the previous solution. This acceptance criterion can lead to converge to local (not global) optima, where the optimization is stagnated. As local optima are consequence of operators (neighborhood) and acceptance criterion, many S-Metaheuristics have been designed to overcome this issue by relaxing the acceptance criterion, and considering some worse solutions as new references. For example, in Random Walk (RW) every new solution (worse or better) is accepted as reference. Other S-Metaheuristics, such as Record-to-Record Travel (RRT), and Great Deluge Algorithms (GDA) use a moderated acceptance criterion. They accept some worse solutions taking into account the quality of the new solution, and some other aspects and parameters. For instance, RRT accepts worse solutions which are not much worse than the best solution in a certain parameter (Deviation). The parameter Deviation directly affects the performance of RRT, because it controls the balance between exploitation and exploration. For example, RRT with a very high value of deviation is similar to RW. In contrast to the modification of the acceptance

criterion, a different approach to avoid local optima is to modify the neighborhood. This approach is used by Variable Neighborhood Search (VNS). It is important to note that local optima are also consequences of the operators [3]. As the neighborhood changes, a solution that is a local optimum in a neighborhood is not necessarily a local optimum in other neighborhood. The underlying idea is that the best solution at the end of the search may be a global optimum because it has been a local optimum in many neighborhood structures. The operators and the criteria to change them affect the performance of VNS. Unless some general guidelines are available to adjust all these S-Metaheuristics [1, 4], the best values depend on the problems, the operators used, and the current state of the optimization process. This is also emphasized by the NFL theorem [2].

Many predictive measures have been proposed to understand and to predict the performance of metaheuristics, e.g. Fitness Distance Correlation (FDC) [3]. FDC computes the correlation between the fitness (evaluation of the solution using the objective function), and the distance (in terms of operators) between each solution and the global optimum. If FDC is near to -1, both characteristics vary in opposite directions, i.e., solutions near to the global optimum (small distance) are good (high fitness). If FDC= -1 this problem may be easy, but if FDC= 1 it may be hard. As FDC needs the values of all the solutions, it has been normally used to study the performance of metaheuristics in certain controlled type of problems. FDC depends on the operator used to establish the distance. For example, solutions 000 and 101 are at distance 2 in terms of 1-bit mutation, but they are at distance 1 in terms of 2-bit mutations. We are not aware of any proposal of changing the neighborhood based on FDC.

2.2 Agents and Metaheuristics

A relevant characteristic of agents is autonomy. Agents are able to act without direct intervention of humans, and are driven by goals. This means that agents need to be proactive, in the sense that they will take the initiative, i.e., decide by themselves what actions to perform in order to satisfy their goals. For this reason, the analysis of the system goals is very relevant in agent-oriented development. The language i^* has shown to be useful for this purpose because it allows to express the main motivation of the actors that interact in a certain environment, and the dependencies among them to solve a problem. This conducts to software requirements, in terms of what a system must do (late requirements) and why it must be do it (early requirements) [5]. Early requirements elicit the dependencies among the different actors (human or software) in order to know why tasks must be done. Late requirements specify which goals can be delegated to software if an agent is able to perform a task with positive influence to satisfy the goal.

Some agent based models of metaheuristics have been developed [6-10]. Most of them rely on the social behavior characteristic of the agent paradigm, by defining agents that exchange messages to guide the search towards the most promising regions. However, no proactive metaheuristics have been proposed, as a consequence of the absence of the explicit modeling of these goals. Particularly, the acceptance criterion and the neighborhood structure of S-Metaheuristics have not been adjusted using an agent based approach. Some mechanisms for parameter adjustment have been

proposed [1, 4], but they are not based on the explicit modeling of the goals, nor the evolution of S-Metaheuristics have been regarded as an environment to be observed and over which an agent may proactively act in order to reach the human goals. A special mention deserves [11]. Although it does not model the goals, it shows how to adjust the temperature in SA to get a certain acceptance probability. It implicitly tries to satisfy a certain objective by controlling the parameters. In the next section we show how the use of i^* contributes to elicit the goals of the different components of the metaheuristics, therefore guiding towards a solution for proactive metaheuristics.

3 Proactive S-Metaheuristics

3.1 Modeling of S-Metaheuristics with i^*

Figure 1 presents the i^* model of the early requirements of an S-Metaheuristic, e.g. RRT. The model shows three relevant actors (circles): Metaheuristic, Function, and Optimizer. Optimizer is a human user that depends on the Metaheuristic (software) to find a solution with the best (biggest) value, using a limited amount of function evaluations. Metaheuristic depends on Function (software) to know the evaluation of each solution. Metaheuristic can do some actions, such as: generation of initial solutions, generation of new solutions by modifications, selection of the solution to be used as reference to generate others, etc. Metaheuristics uses the parameters and operators set by Optimizer. The goal of Optimizer is to avoid stagnation in local optima.

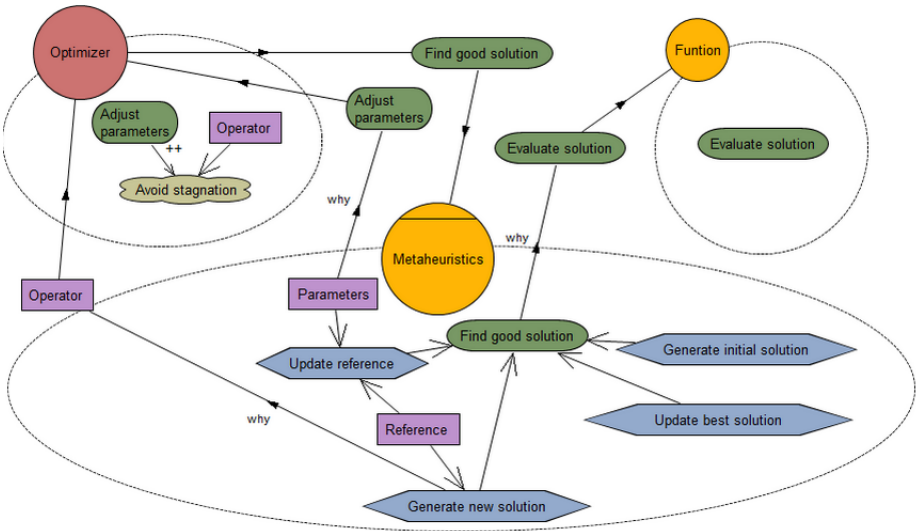


Fig. 1. Early requirements: Strategic Rationale (SR) model of Metaheuristics and Optimizer

Figure 2 (late requirements) shows how the goal “Find good solution” may be passed from Optimizer to the agent Metaheuristic. This is done by allowing Metaheuristic to adjust the operators and the parameters (task or plans). This task positively

contributes to the goal “Find good solution”. The parameters, operators, and the last solutions generated can be seen as resources. The optimization is an environment where the agent Metaheuristic is situated, and which evolves according to the taken actions. As the intention behind the parameter adjustment is to avoid stagnation in local optima, it is important to give a plan to the agent in order to do so. The next section explains these plans in the context of two S-Metaheuristics: LS and RRT.

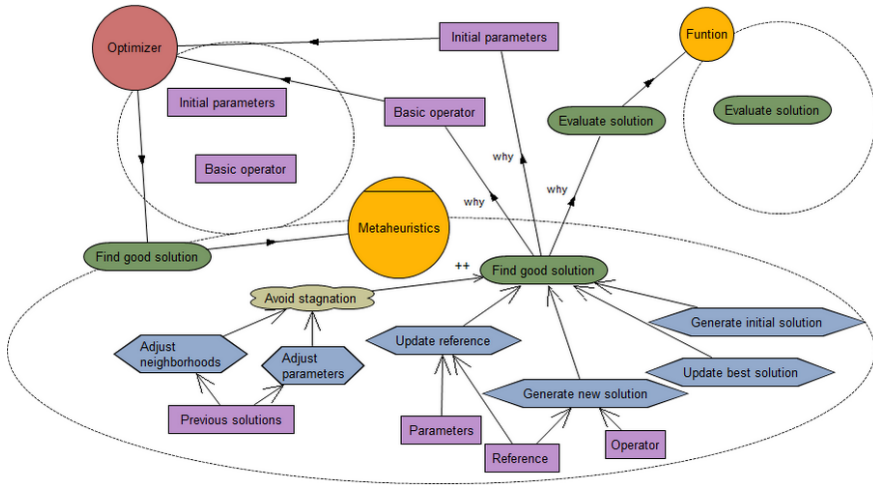


Fig. 2. Late requirements: Strategic Rationale (SR) model of Metaheuristics and Optimizer

3.2 Plan for Proactive Adjustment of Parameter Deviation of RRT

In RRT, deviation D must have a value that allows the acceptance of at least one solution in a certain time window (TW), e.g. $TW=100$ last solutions. Despite the next solutions are unknown, it is possible to estimate good parameters setting using the solutions generated in the near past. If E_{best} is the evaluation of the best solution found in a maximization problem, a new solution X will be rejected if $E_x < E_{best} - D$. Suppose that E_1 is the evaluation of the best solution generated during a time window, and E_2 is the evaluation of the second best solution in this time window. If D is the average of E_1 and E_2 , then the former will be accepted and the later will be rejected. In consequence, the solutions of the time window serve to proactively adjust the parameter D for the next time window, in order to expect that at least one solution will be accepted, and the stagnation in local optima is avoided. The values of D must be adjusted after every time window, in order to adapt it to the state of the optimization. Thus, this plan for adjustment of deviation is triggered after each time window.

3.3 Plan for Proactive Adjustment of Neighborhoods Based on FDC

A neighborhood structure is a graph where each node is a solution, and each edge between two solutions implies that it is possible to obtain one solution by the application of an operator to the other. The existence of local optima is a consequence of the

neighborhood structure [3]. Based on a basic operator (such as 1-bit mutation) given by the optimizer it is possible to define some derived operators by the repeated application of the basic operator. For example, 2-bit mutation operator may be defined as the application of 1-bit mutation twice. Thus, many neighborhood structures may be used (as in VNS) without the explicit definition of these operators by the Optimizer. As FDC is a measure of problem difficulty that depends on the operator used, it may be used to know which neighborhood structure is the best. FDC needs a complete exploration of the search space, what is impractical for optimization of real problems but, it is possible to get an estimation of FDC based on the previous solutions.

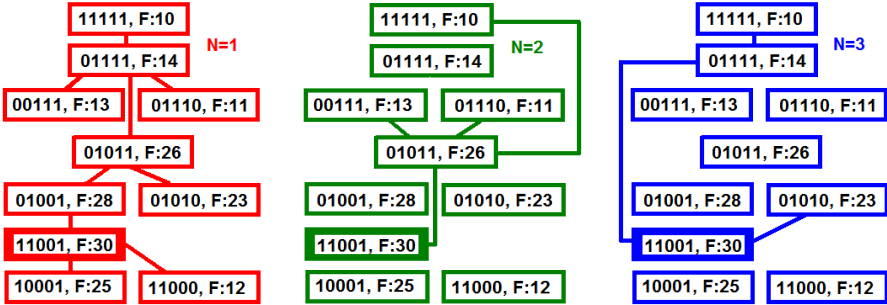


Fig. 3. Neighborhoods using N-bit mutation, for N=1, N=2 and N=3

As it is known which solution was used as the reference to generate a new one, then it can be known that both solutions are neighbors. A graph with the last solutions generated and an edge between all known neighbors is a sample of the neighborhood structure. The best solution in this set may be used as a pseudo-optimum. This information can be used to get an approximate value of FDC. The graph in red, on the left part of Figure 3 is an example of the information that can be used to obtain an approximation of FDC using ten hypothetical solutions. As 30 (i.e solution 11001, with fitness F=30) can be used as pseudo-optimum, the distance from every solution to 30 can be calculated. Suppose that these distances are in terms of the basic operator (1-bit mutation, in red). As a 2-bit mutation can be expressed by applying a 1-bit mutation twice, the same graph can be used to obtain the graph of the neighborhood in terms of 2-bit mutation (in green color, at the center of Figure 3), where two solution are neighbors if they are at distance 2 in the graph of the basic operator. The same analysis can be used to obtain the graph for a 3-bit mutation operator (in blue, at right in Figure 3). The distances from every solution to the pseudo-optimum can be obtained using each graph (operator). In some cases, it is impossible to calculate the distance, e.g. there is not a path from 14 to 30 in the graph for N=2 (2-bit mutation). In this example, for the 2-bit mutation operator FDC= -0.96. As it is close to -1, it suggests that it is preferable to use the 2-bit mutation operator (for 1-bit mutation FDC=-0.73, and for 3-bit mutation FDC=-0.83). It is important to note that the Optimizer has just defined the basic operator, but the Metaheuristic can proactively use the best operator. In summary, the plan for adjustment of the neighborhood consists of selecting the best operator (basic or derived) based on FDC. After each time window, Metaheuristic triggers this plan to update this analysis and to decide the operator for the next time window.

3.4 Proactive Metaheuristics with Adjustment of Deviation and Neighborhood

The plans for adjustment of deviation and neighborhood may be used in isolation. If we only use the plan for proactive adjustment of neighborhood, this may be considered a Proactive Local Search (PLS) with adjustment of neighborhood. The same can be done with the plan for adjustment of parameter deviation, which may be considered a Proactive RRT (PRRT).

Besides, both plans may be used in combination in order to get a robust performance. A simple way of doing so may be to allow both plans to act independently in competition. After a certain time, the one with best performance is allowed to be used for the rest of the time. As function evaluation is a limited resource, it is important to use it efficiently. The amount of fitness evaluations must be divided between the elaboration of the proposed solution by each metaheuristic, and the refinement of the solution by the winner. Here two versions are explored. In PRRT*PLS, no time for refinement is allowed, i.e. the total amount of evaluations is divided between PRRT and PLS, and the best proposal of them is returned. In PRRT*PLS+R the refinement is included, i.e. PRRT and PLS run for a quarter of the total amount of fitness evaluations. Then, the best are allowed to run for the half remaining of fitness evaluation.

4 Experiment and Discussion

4.1 Experiments: Functions and Metaheuristic in Comparison

This paper shows the experimentation with 28 100-bit binary coded functions. Each function consists of 25 copies of 4-bit building blocks and has an optimum value of 100. Each building block for the 28 functions is a unitation-based function. Four of them have been used before [3, 6, 12]: Deceptive, Onemax, RoyalRoad, and Plateau. The unitation of a bit string is the number of "1" inside the string. The difference among the 28 functions is based on different building block functions. For each possible value of unitation in a block, each building block function returns a value of 0, 1, 2, 3, or 4. As in [12] every building block function only returns 4 if the value of unitation is 4. This implies that the only global optimum is the solution with 100 bits with 1 (every 4-bits block is 1111, with perfect unification value of 1). The different values that each building-block function returns for the rest of possible value of unification imply different building block functions. In the four building block functions of [12], only the values of 0, 1, 2, or 3 are used. As it can be seen, with the same structure of these four functions, it is possible to define 44=256 blocks functions. Deceptive is supposed to be hard (FDC= 1), and Onemax is supposed to be easy (FDC= -1). We generate other 24 build block function with different values of FDC, in order to get a best sample of the 256 possible block functions. The 28 building block functions are shown in Table 1. Functions f0123, f0003, f0000, are f3210 are Onemax, Plateau, RoyalRoad, and Deceptive, respectively.

Table 1 also summarizes the 28 building block functions by using average (Ave), median (ME), minimum (MIN), maximum (MAX) and the standard deviation (STDEV) of some measures (BFDC, AveB, X-Y). BFDC measure computes FDC for

each building block function using the possible unitation values. AveB computes the average of the five values that function returns for each value of unitation. The measures X-Y (e.g. 0-1, 0-2, 0-3, 1-2, 1-3, and 2-3), compute the differences between the values that each function returns for the value of unitation X, and Y. For instance, in f3210 (Deceptive) the difference between the value assigned for the unitation value of 1 (2) respect to the unitation value of 3 (0) is equal to 2 ($2=2-0$). These 28 functions have a good diversity in all these measures, as it can be observed in Table 1.

Table 1. Measures on the block functions that have been used in the experiment

Building block functions				Measures	Ave	ME	MIN	MAX	STDEV
f0000	f0123	f1120	f3000	0-1	0.75	0	-1	3	1.43
f0001	f0131	f1221	f3001	0-2	0.39	0	-3	3	1.75
f0003	f1001	f2000	f3002	0-3	0.21	0	-3	3	1.71
f0022	f1002	f2001	f3010	1-2	-0.36	0	-2	1	0.73
f0111	f1012	f2012	f3012	1-3	-0.54	-1	-3	2	1.37
f0120	f1111	f2200	f3102	2-3	-0.18	0	-3	2	1.39
f0122	f1112	f2220	f3210	BFDC	-0.03	0	-1	1	0.65
				AveB	1.02	1	0	1.50	0.41

The performance of the PRRT, PLS, PRRT*PLS, and PRRT*PLS+R, are compared to the S-Metaheuristic TA, GDA, RRT, and a classical LS. Two population-based metaheuristics (P-Metaheuristics) [1] are also compared: Evolution Strategies, and Genetic Algorithms. The identifiers and the parameters of all these metaheuristics used in the comparison are:

- LS: Stochastic Local Search (or Hill Climbing), equal solutions are accepted.
- RRT: Record-to-Record-Travel, deviation $D=5$.
- GDA: Great Deluge Algorithm, rain $R=0.01$, initial water level $WL=0$.
- TA: Threshold Accepting, threshold $T=2$.
- GA100: Genetic Algorithms, population size 100, steady-state replacement, truncation selection of the best 50, uniform crossover, 1-bit mutation, probability of crossover: 1, probability of mutation: 1.
- ES5: Evolution Strategies, population size 5, steady-state replacement, truncation selection of the best, 1-bit mutation, probability of mutation: 1.
- ES100: Evolution Strategies, population size 100, steady-state replacement, truncation selection of the best 20, 1-bit mutation, probability of mutation: 1.

These settings come from usual recommendations [1, 4] with manual adjustment. For PRRT, PLS, PRRT*PLS, and PRRT*PLS+R the time window $TW=200$ was used.

4.2 Results and Discussion

For each experiment of the 11 metaheuristics on the 28 test function, 30 independent runs were executed until 10000 fitness evaluations. The best solution found on each

run was registered. These 30 results for each metaheuristic in each function were summarized using the average. In addition, as the Kruskal-Wallis test revealed differences amongst the algorithms in each function, we perform a comparison between each pair of algorithms using the unpaired Wilcoxon test (always with significance of 0.05). When a metaheuristic is better than another in the Wilcoxon test, the winner receives +1 and the loser receive -1. So, if one metaheuristic outperforms all the others in a function it gets a perfect score $SW=10$. In general, these SW values are integers between -10 and 10, and was also used to summarize the performance of the algorithm in each function. The importance of non-parametric tests (e.g. Wilcoxon test, Friedman test, etc.) for comparing metaheuristics has been demonstrated in [13].

Table 2. Summary of the performance in the 28 block functions

	OM	PI	RR	De	AA	ASW	SR	SWp
PRRT*PLS+R	100	100	99.7	82.9	93.3	4.4	232,5	7
PRRT*PLS	100	100	100	83.1	92.9	4.3	232,0	6
PRRT	100	100	100	81.0	91.2	3.0	220,0	4
PLS	100	100	98.9	82.5	91.4	1.5	178,0	1
ES100	98.2	97.4	85.1	83.4	90.5	0.6	174,0	1
GA100	99.8	99.1	86.9	83.6	90.7	-0.2	158,5	2
ES5	100	100	98.9	83.2	88.6	-0.1	162,5	-1
TA	66.4	87.9	100	100	80.8	-1.6	133,5	0
LS	100	100	100	78.0	83.2	-2.5	151,0	-7
GDA	99.7	99.6	79.7	83.2	82.4	-4.0	114,0	-6
RRT	79.9	86.7	77.3	80.7	78.5	-5.5	92,0	-7

Table 2 shows the average performance in the four original block functions: Onemax (OM), Plateau (PI), RoyalRoad (RR), and Deceptive (De). Due to space limitations, we can not show the particular results in the rest 24 functions. The last columns summarize the results in all functions. The average of all averages is shown in column AA. The column ASW shows the average of all the SW values of each metaheuristics. Finally, we compare all the algorithms using the Friedman test based on the 28 averages which results in a significant difference ($p\text{-value}=5*10^{-11}$). The sum of ranks in Friedman test is shown in column SR. Finally, we perform a pair-wise comparison between all the algorithm (using the paired Wilcoxon test with significance of 0.05), and the sum of the results (using the same procedure presented before for SW) is presented in column SWp. The results clearly show that four proactive metaheuristics perform well, with the advantages that no parameter tuning is necessary. They outperform other S-Metaheuristics and P-Metaheuristics. Another interesting result is that this setting of TA performs perfectly in Deceptive, but badly in Onemax. In general, results confirm that proactivity is a promising line of research in metaheuristics.

5 Conclusions

The paper has introduced proactive versions of the *S*-Metaheuristics LS and RRT based on the agent paradigm. The proactive metaheuristics use the information of the environment (solution previously evaluated) in order to take actions (adjustment of parameters and the neighborhood). These actions are oriented towards the goal of the human optimizer (avoiding local optima). This approach comes from a social model of the metaheuristics based on *i**framework, that derives early requirements that conduct to the proactive behavior. Results on 28 100-bit binary coded functions show that proactive metaheuristics obtain good results in comparison to other metaheuristics.

Acknowledgments. This work has been supported by the project *SociAAL* (TIN2011-28335-C02-01), funded by the Spanish Ministry for Economy and Competitiveness.

References

1. Talbi, E.G.: *Metaheuristics: From Design to Implementation*. John Wiley & Sons (2009)
2. Wolpert, D.H., Macready, W.G.: No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation* 1(1), 67–82 (1996)
3. Jones, T., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: 6th Int. Conf. on Genetic Algorithms, Pittsburgh, pp. 184–192 (1995)
4. Birattari, M.: *Tuning Metaheuristics*. SCI, vol. 197. Springer, Heidelberg (2009)
5. Yu, E.S.: Social Modeling and *i**. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) *Conceptual Modeling: Foundations and Applications*. LNCS, vol. 5600, pp. 99–121. Springer, Heidelberg (2009)
6. González, J.R., Cruz, C., del Amo, I.G., Pelta, D.A.: An adaptive multiagent strategy for solving combinatorial dynamic optimization problems. In: Pelta, D.A., Krasnogor, N., Dumitrescu, D., Chira, C., Lung, R. (eds.) *NICSO 2011*. SCI, vol. 387, pp. 41–55. Springer, Heidelberg (2011)
7. Lepagnot, J., Nakib, A., Oulhadj, H., Siarry, P.: A New multiagent Algorithm for Dynamic Continuous optimization. *Int. Journal of Applied Metaheuristic Computing* 1(1), 16–38 (2010)
8. Aydin, M.E.: Coordinating metaheuristic agents with swarm intelligence. *Journal of Intelligent Manufacturing* 23(4), 991–999 (2012)
9. Malek, R.: Collaboration of Metaheuristics Algorithms through a Multi-Agent System. In: Mařík, V., Strasser, T., Zoitl, A. (eds.) *HoloMAS 2009*. LNCS, vol. 5696, pp. 72–81. Springer, Heidelberg (2009)
10. Li, B., Yu, H., Shen, Z., Miao, C.: Evolutionary Organizational Search. In: 8th Int. Conf. on Autonomous Agents and Multiagent Systems, Budapest, pp. 1329–1330 (2009)
11. Poupaert, E., Deville, Y.: Simulated Annealing with estimated temperature. *AI Communications* 13(1), 19–26 (2000)
12. Wang, H., Wang, D., Yang, S.: A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems. *Soft Computing- A Fusion of Foundations, Methodologies and Applications* 13(8-9), 763–780 (2009)
13. García, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study. *Journal of Heuristics* 15(6), 617–644 (2009)

An Ontological and Agent-Oriented Modeling Approach for the Specification of Intelligent Ambient Assisted Living Systems for Parkinson Patients

Iván García-Magarino¹ and Jorge J. Gómez-Sanz²

¹ Departamento de Ingeniería Informática y Organización Industrial,
Facultad de Enseñanzas Técnicas,
Universidad a Distancia de Madrid,
Collado Villalba, Madrid, Spain
ivan.garcia-magarino@udima.es

² Departamento de Ingeniería del Software e Inteligencia Artificial,
Facultad de Informática
Universidad Complutense de Madrid,
Madrid, Spain
jjgomez@fdi.ucm.es

Abstract. The goal of this work is to introduce requirements analysis for Ambient Assisted Living (AAL) applications for patients with Parkinson Disease (PD). The work is part of SociAAL project, which aims to reduce costs in the development of AAL applications for PD patients. A first step is the understanding of the needs of the PD patient and the differences with other possible AAL users. This paper uses a hybrid approach in which an ontology and agent-oriented models are used to formalize initial application requirements. This hybrid approach is being essayed with transcriptions of interviews made along the SociAAL project.

Keywords: Agent-oriented software engineering, assisted living, modeling, multi-agent system, ontology, Parkinson disease.

1 Introduction

Ambient Assisted Living (AAL) applications are mainly aimed at improving the quality of living of elderly. Parkinson Disease (PD) is an illness with a high impact in the elderly. However, AAL applications cannot be said to be widely effective in dealing with the issues of PD patients. PD patients have problems controlling movements in general. A very specific affection is bradykinesia, or immobility. A PD patient can stay still in the middle of the room unable to move and without falling. They need an external stimulus to keep moving, like challenging the patient to stomp the foot of the caregiver or playing some kind of music. Reader can find a more detailed description of PD in our previous analysis [1].

Developing specific AAL applications for PD implies focusing in a very specific segment of the elderly and investing time and effort in creating ad-hoc solutions, or extensions to existing ones. Though, medically, PD is well known, from the perspective of AAL has not been researched as much. Hence, the risk of creating the wrong aiding system is high.

The SociAAL project aims to reduce the development cost of AAL applications for PD patients and one of the involved steps consists in formalizing the needs PD patients have. This may be useful for others as a better starting point for creating generic, perhaps commercial, AAL systems that do answer the needs of this collective. For this aim, a field study has been conducted in a group of PD patients and the project is proceeding to analyze and formalize the collected information.

The SociAAL project intends to apply Model Driven Development (MDD) solutions to transform successively initial requirements into the final AAL system. Concretely, this analysis involves the use of an ontology and agent-oriented models. The ontology is mainly used for the elicitation of the requirements of a AAL application through an extensive categorization of the patient and related people, including the status and particularities of his/her PD, his/her circumstances as caretakers, and other relevant issues. In addition, the ontology also represents the features of the AAL application. Some basic rules are provided for validating that the AAL application can really assist a given patient. The information is also regarded from the perspective of Multi-Agent Systems (MAS) by means of the INGENIAS methodology [5], looking for a formalism that permits to capture pertinent behavioral information and run preliminary simulations that validate the captured information. The result is a MAS that represents the actors of the analysis. The later also serves to start working in the solution, which would be a new system of agents interacting with those identified during this process.

Our previous work [4] introduced a metamodel for guiding the requirements capture for PD patients. However, we discarded using a Modeling Language (ML) for capturing the PD patient condition and status. Though MLs can use means, like constraint languages, to validate a model correctness, we found more advanced and better supported the facilities traditionally associated to ontological approaches, such as validation and reasoning solutions. At the same time, expressing the behavior of the PD patients, related actors, and their interplay, was still easier with a ML, specially if the ML was designed for this aim. This is the case of a MAS ML, and more specifically, the INGENIAS MAS ML, which follows a Believes-Desires-Intentions (BDI) stance. BDI can translate intuitively actions of the actors and the reasons behind them.

Therefore, the current work tries to study again the problem using a hybrid approach. It applies an ontology to capture the PD patient situation and then explores connections between this ontology and the elements captured using a MAS ML. In addition, the current work is now based on the knowledge concerning recent transcriptions of a field study with PD patients and professionals as part of SociAAL activities [1]. An interview can have 120 questions and 9000 words, for instance. This paper discusses some experimental results with these these interviews, and shows a case study.

The remainder of this article is organized as follows. The next section introduces the most relevant works related to the current one, indicating the improvements or differences. Section 3 presents the hybrid approach determining the most relevant aspects of the ontology and the agent-oriented models. Section 4 presents the experimentation of the current approach with a real case of a PD patient. Finally, section 5 mentions the conclusions and future work.

2 Background

There are several works that present ontologies for patients of PD or other similar diseases. For instance, Gupta et al. [6] presents a way of formalizing ontologies for neurological disorders, such as Alzheimers disease, PD and schizophrenia. They present disease maps that construct knowledge-base with the relations between concepts by means of ontologies, and this approach is integrated within the work of a consortium for large-scale data and computational grid around neuroimaging data. In addition, Riaño et al. [7] present an ontology for customizing the health-care of chronically ill patients. This ontology is mainly aided at supporting the decisions on how take care of these patients. This approach is specially useful for checking difficult decisions that require the consensus of a group of professionals. Sometimes, some professionals miss some relevant information in some cases, providing a regrettable decision. Furthermore, Bickmore et al. [2] uses a behavioral medicine ontology in a reusable framework for constructing dialogue systems for advising patients through the web or phone. Nevertheless, none of the ontologies consider the information of AAL applications for constructing these and validating their properties, as the current approach does.

Some works combine the ontologies and MASs conforming hybrid intelligence approaches. For instance, the supply chain management has been performed with a MAS for guiding the negotiation protocols and an ontology for conforming the negotiation knowledge [8]. In this manner, the negotiation agents are more adaptive for different negotiation environments. In addition, the secure type-2 fuzzy ontology multi-agent platform (ST2FO-MAS) [3] automates the process of manual air ticket booking. The ontology addresses the information management of ticket booking and maintains the security, while the MAS improves the performance thanks to its autonomous working scheme. The current approach also uses a hybrid intelligence approach with ontologies and MASs. However, the current work has a completely different aim, which is the specification of PD patients for developing AAL MASs with agent-oriented models.

3 An Ontological and Metamodeling Approach for Determining the Requirements of AAL MASs

The current approach combines the use of an ontology and agent-oriented models for the specification of AAL MASs for PD patients. The ontology is mainly used for the elicitation of the requirements of a AAL application through a categorization of the patient features and the AAL system. Then, the main relevant

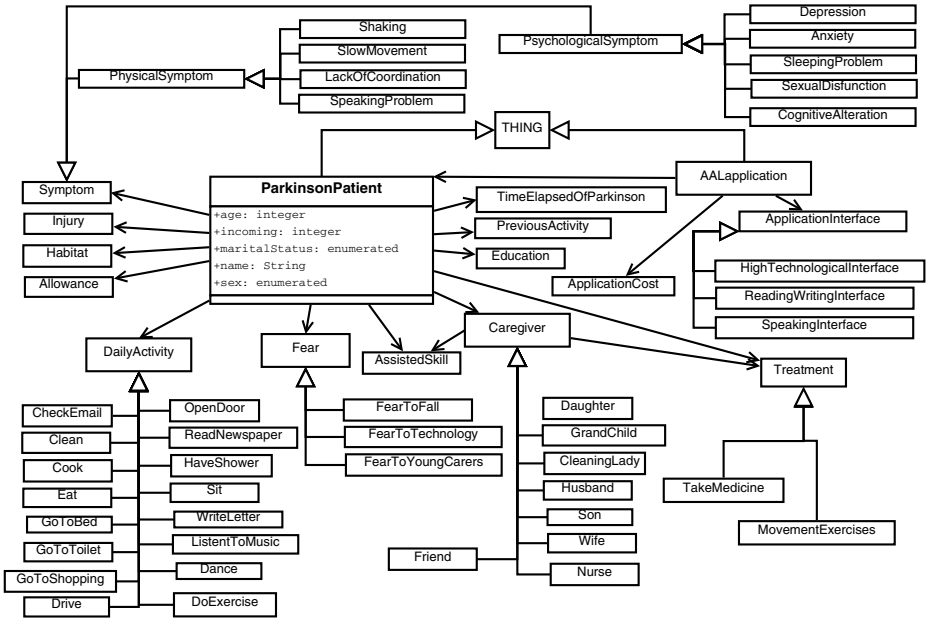


Fig. 1. Excerpt of the ontology for PD patients

aspects are transferred to agent-oriented models. From this point forward, the skeleton of a AAL MAS can be generated from this model with the INGENIAS methodology and the IDK tool, following a MDD approach.

3.1 Ontology for PD Patients

The add-value of ontologies respect to metamodels is that they can directly express consistency rules, in order to know whether a specific group of instances of the concepts of an ontology are coherent. This is special valuable when determining whether a PD patient can really use an AAL application. For instance, if a patient has fear of technological devices, an application with a technological device would not be really useful. Similarly, a patient with a speaking problem will not be able to use an AAL application with a speaking interface.

The presented ontology is composed of 79 classes, and the most relevant 59 classes of these are presented in Figure 1. Some of these classes are directly subclasses of the Thing general class. The main classes of this ontology are the ParkinsonPatient and the AALApplication. The former represents a PD patient for who the development of an AAL application is intended, while the latter represents an AAL application.

The ParkinsonPatient class is related to other classes that describe different aspects of the patient. To begin with, there are several classes that describe the past history of the patient, which can be useful for adjusting the application to the specific past experience of the patient. The PreviousActivity class mainly

refers to the jobs or hobbies that a patient had before their PD was detected. In case the patient did not have a job, he/she can be dedicated to the domestic tasks. The Education class indicates the level of education, which can be for instance a grade or a PhD. The TimeElapsedOfParkinson class indicates the number of years since the PD was detected in the patient, which for instance can be useful to determine the level of the disease and consequently the grade of assistance that the AAL application can offer to this particular patient.

There is a group of classes that are related to the features of a patient. The Symptom class indicates the symptoms of the PD in the patient determining its level with a natural member from one to five. These symptoms are represented with a hierarchy of classes for the different known symptoms of PD, classifying these in physical and psychological ones. The DailyActivity class refers to the activities that the patient performs daily. These activities are defined with subclasses of the mentioned class, and some examples of these are presented in Figure 1. According to the symptoms and level of the disease, the patient needs to be assisted in certain skills for performing his/her daily activities, and these assisted skills are represented with the AssistedSkill class and its subclasses. The Fear class determines the fears that a patient can have. These fears can be obstacles for performing the daily tasks or interacting with an AAL application. Figure 1 shows the Fear class and some of its subclasses. For example, the FearToTechnology subclass is related to the fact that a patient is not used to technological devices and is afraid of these; hence, this will influence in the AAL application interface. Besides the PD, a patient can also have other injuries, which for instance can also hinder his/her mobility. These injuries are represented with the Injury class and its subclasses, which only cover some general common injuries. These subclasses are omitted and an example of these is the KneeInjury subclass.

Another set of classes represent the circumstances surrounding a patient. In particular, the Habitat class determines the number of members in the patient home and the population of the town or city where he/she lives. The Caregiver class can determine the features of each person that takes care of the patient. The human carers can be classified according to their relation with the patient (either related by blood, professional or by friendship). Figure 1 shows the Caregiver class and its subclasses, such as wife, husband, daughter, cleaning lady, nurse and friend. The Allowance class determines the amount of money the patient is receiving from the State if this is the case.

Other basic properties of patients are represented as slots of the Parkinson-Patient class, as one can observe in Figure 1. Some of these slots refer to his/her name, age, sex, marital status and incoming. The slots of the other classes are omitted in the diagram for the sake of simplicity in the presentation.

Moreover, there is a group of classes that are related to the AAL application. The AALapplication class represents the AAL application that is going to be developed for assisting a certain PD patient. This application is related to the ApplicationCost class, which contains the specific cost and financial conditions.

In addition, the `ApplicationInterface` determines the kind of interface of an AAL application, and is shown with its subclasses in Figure 1.

Finally, there are several rules expressed with the Semantic Web Rule Language (SWRL) for validating some aspects in the instances of the ontology. For example, one of the subclasses of `ApplicationInterface` is the `HighTechnologicalInterface` class. This class is useful for checking that an AAL application is not recommended for a patient who is afraid of technology (represented with the `FearToTechnology` class), by means of Rule 1. Another subclass of `ApplicationInterface` is the `SpeakingInterface` class, which Rule 2 uses to determine that this kind of interface is not appropriate for a patient with a speaking problem (represented with the `SpeakingProblem` subclass of the `PhysicalSymptom` subclass of the `Symptom` class).

Rule 1:

$$\begin{aligned} & ParkinsonPatient(?p) \wedge hasFear(?p, ?f) \wedge FearToTechnology(?f) \\ & \wedge AALapplication(?a) \wedge hasApplicationInterface(?a, ?i) \\ & \wedge HighTechnologicalInterface(?i) \longrightarrow isNotAppropriateFor(?a, ?p) \end{aligned}$$

Rule 2:

$$\begin{aligned} & ParkinsonPatient(?p) \wedge hasSymptom(?p, ?s) \wedge SpeakingProblem(?s) \\ & \wedge AALapplication(?a) \wedge hasApplicationInterface(?a, ?i) \\ & \wedge SpeakingInterface(?i) \longrightarrow isNotAppropriateFor(?a, ?p) \end{aligned}$$

3.2 Agent-Oriented Modeling of PD Patients

The analysis of the different interviews shows an expected sequence of activities of daily living. Since SociAAL is about the social aspects of PD, information about caregivers was taken as well, and the interactions between caregivers and caretakers.

A first analysis conformed an ontology of relevant concepts in the PD. These concepts were repeatedly found in the first three interviews. In addition, there was an analysis of the behavior implied from the daily living activities of both caregivers and caretakers. The analysis shows that there is a dependency between the activities of one and the other. Studies such as [9] show most of the caregivers are the wife or the husband of the patient. When other relatives are considered, the interaction gets more complex, involving non-directly PD related activities of daily living with others strongly related.

To understand this interaction, it is important to realize the motivation behind. The Agent Oriented Paradigm is an appropriate mechanism to evaluate this behavior and capture this interaction. This work proposes to apply the BDI paradigm for the agent-oriented modeling of PD patients with their needs, as a starting point for the development of a customized MAS. In particular, the current approach mainly focuses on relationships such as the following one. One activity is likely to fail in a dangerous way. Such tasks require the caregiver to be specially aware and synchronize activities, like in the following examples:

- PD patient eats. Then, the caregiver watches the PD patient to eat and that the food is properly engulfed, since PD patients may lose the swallowing reflex. Caregiver may eat too.
- PD patients show fear of falling or getting stuck in a position, for instance. Then, the caregiver may need to watch or pay attention from time to time if the caretaker moves or if nothing is heard from the caretaker.

Therefore, the caregiver must synchronize his/her activities to be compatible with the caretaker. One may realize that the closer the person is, the closer the dependency is. If the caregiver is a capable person, it may seem negative to completely dedicate his/her time to the caretaker. There could be other caregivers who coordinate to provide assistance. Some can be dedicated full time to the patient's care. Others need a relief from time to time. This fact implies that there could be additional actors involved and their interplay could be important.

Moreover, actors may find compelled to provide assistance, even though it affects negatively to their own individuality. This constitutes a frequent conflicting goals situation, but there are also positive ones, like the reward they perceive when helping the loved ones.

In any case, the current approach recommends including other caregivers and their motivation, specially reinforcing and conflicting goals, in the agent-oriented models, since it amplifies the scope of the study to address other activities of daily living, not necessarily related to the PD patient. Representing both, it is possible to understand better the situation of the caregiver and the caretaker.

4 Case Study

The current approach has been applied to determine the customization of AAL applications of three real PD patients, according to some interviews made in SociAAL. This customization of the current approach takes more aspects into account when comparing to other similar approaches that use ontologies for health-care systems [6,7]. In particular, the current approach has the advantage of modeling and validating the appropriate system interface considering the cognitive dysfunctions and fears of patients.

As a case study, the current approach has been applied for one of these patients, who is labeled as E12 and whose private information is omitted. This patient has been represented with an instance of the presented ontology, as one can observe in Figure 2. Firstly, the ParkinsonPatient class is instantiated with the E12 instance. Her two caregivers are represented with instances of the Husband and Nurse subclasses of the Caregiver class. The diagram collects the most relevant activities in which the patient is assisted, with instances of the AssistedSkill class. These activities are (1) to open cans or coffee maker and (2) to recognize new symptoms of PD. Her most relevant daily activities are represented with instances of some subclasses of the DailyActivity class, such as cooking meals, listening to rock music, driving a car and dancing. Her most relevant symptom, which is motor disability, is represented with an instance of

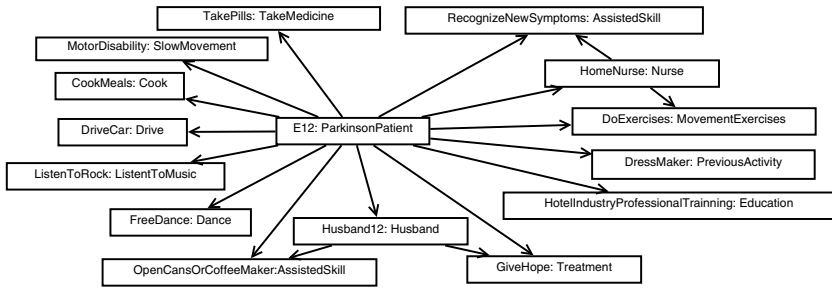


Fig. 2. Instance of the ontology for the E12 patient

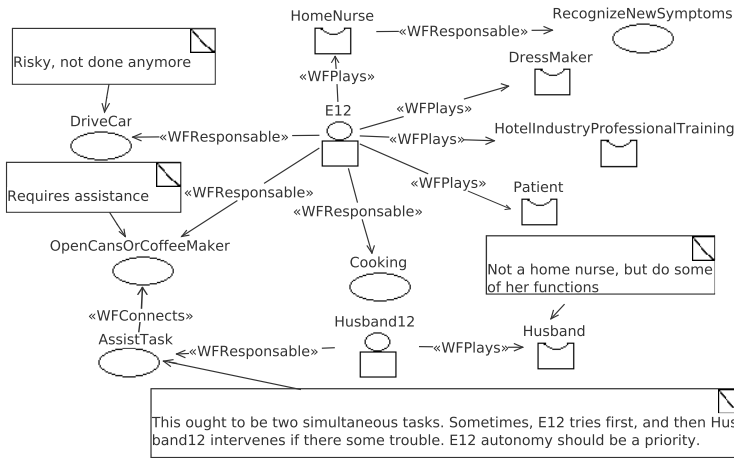


Fig. 3. Actors in an agent-oriented model for the E12 patient

the SlowMovement class. The most important treatments are to be given hope, to do exercises and to take pills, represented respectively with instances of the Treatment class and its MovementExercises and TakeMedicine subclasses. Other aspects such as her education are also represented.

After this, several agent-oriented models have been defined based on the aforementioned ontology instances and additional information contained within the interview. In particular, Figure 3 shows the actors with a task and goals model with the INGENIAS notation for this patient. Most part of the information of the ontology instance is transferred to this model. The main actors are the patient and her husband, and are represented as agents. These agents play several roles that represent her previous activities and the assistance on behalf of the husband. These agents are associated with tasks that determine her daily activities.

Another task and goal diagram is defined to identify the goals for the E12 patient, as one can observe in Figure 4. These goals do influence each other and are

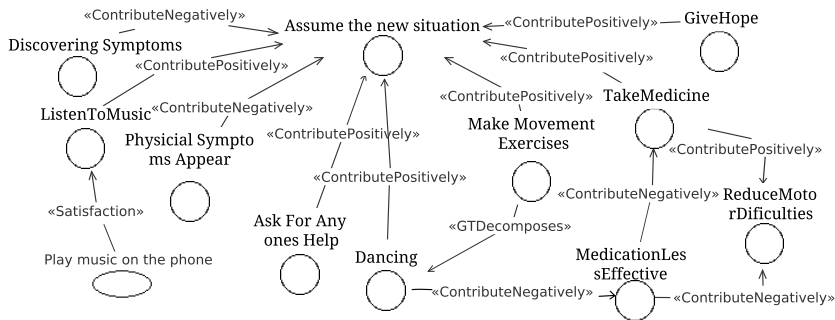


Fig. 4. Goals identified for the E12 patient in an agent-oriented model

later assigned to tasks (whose execution permits to attain goals), agents (to express particular motivation), or roles (to represent generic motivation associated to a concrete role). These goals determine the intentions of some daily activities, which are treatments in many cases, such as take medicine, do exercises, or be given hope. In other cases, some daily activities are not strictly treatments but contribute positively to the PD, and are also represented with goals. The main goal of the customized MAS is that the patient accepts the disease and tries to live with it. All the other goals are established in hierarchy, indicating which ones contribute positively to this main goal and which ones contribute negatively. The goals can be satisfied with specific tasks of the customized AAL MAS, such as the one related to including music on the phone. Executing those tasks associated to concrete goals, we can foresee which other aspects will be affected negatively and try to compensate accordingly, if necessary.

After modeling the problem with agents, INGENIAS tools are used to compile the specification and detect flaws. The compilation process implies ensuring basic consistency rules, like, 'there cannot be a goal without a task' or 'a more thorough task description is required'. This led to further expanding Figures 3 and 4 until an executable description was obtained. The resulting specification is more complex, but it serves to start working in the AAL solution. This solution would be modeled as another agent which interacted with those in Figure 3 and help attaining their goals.

5 Conclusions and Future Work

A hybrid approach is presented for capturing requirements for AAL systems oriented towards PD patients. It starts from the examination of the PD patients, their caregivers, and other circumstances. All this information is collected in an instance of the ontology. The ontology can infer or validate certain requirements of AAL applications. Then, the most relevant information is transferred to BDI-based agent-oriented models. These models are the starting point of the MDD of AAL MASs customized for certain PD patients. The ontology is based on

the interviews with three actual PD patients with their closest relatives, and the approach is applied in a patient as a case study. As future work, a specific ML, probably an agent-oriented one, is expected. It will be used for customizing MASs for particular PD patients.

Acknowledgements. This work has been done in the context of the project *Social Ambient Assisting Living - Methods* (SociAAL), supported by Spanish Ministry for Economy and Competitiveness, with grant TIN2011-28335-C02-01.

References

1. Arroyo, M., Finkel, L., Gomez-Sanz, J.J.: Requirements for an intelligent ambient assisted living application for parkinson patients. In: Corchado, J.M., Bajo, J., Kozlak, J., Pawlewski, P., Molina, J.M., Julian, V., Silveira, R.A., Unland, R., Giroux, S. (eds.) PAAMS 2013. CCIS, vol. 365, pp. 441–452. Springer, Heidelberg (2013)
2. Bickmore, T.W., Schulman, D., Sidner, C.L.: A reusable framework for health counseling dialogue systems based on a behavioral medicine ontology. *Journal of Biomedical Informatics* 44(2), 183–197 (2011)
3. Bukhari, A.C., Kim, Y.-G.: Integration of a secure type-2 fuzzy ontology with a multi-agent platform: a proposal to automate the personalized flight ticket booking domain. *Information Sciences* 198, 24–47 (2012)
4. García-Magariño, I.: Defining and transforming models of parkinson patients in the development of assisted-living multi-agent systems with INGENIAS. In: Corchado, J.M., et al. (eds.) PAAMS 2013. CCIS, vol. 365, pp. 460–471. Springer, Heidelberg (2013)
5. Gomez-Sanz, J.J., Fuentes, R., Pavón, J., García-Magariño, I.: Ingenias development kit: a visual multi-agent system development environment. In: *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Demo Papers*, pp. 1675–1676. International Foundation for Autonomous Agents and Multiagent Systems (2008)
6. Gupta, A., Ludäscher, B., Grethe, J.S., Martone, M.E.: Towards a formalization of disease-specific ontologies for neuroinformatics. *Neural Networks* 16(9), 1277–1292 (2003)
7. Riaño, D., Real, F., López-Vallverdú, J.A., Campana, F., Ercolani, S., Mecocci, P., Annicchiarico, R., Caltagirone, C.: An ontology-based personalization of healthcare knowledge to support clinical decisions for chronically ill patients. *Journal of Biomedical Informatics* 45(3), 429–446 (2012)
8. Wang, G., Wong, T., Wang, X.: An ontology based approach to organize multi-agent assisted supply chain negotiations. *Computers & Industrial Engineering* 65(1), 2–15 (2013)
9. Williamson, C., Simpson, J., Murray, C.D.: Caregivers’ experiences of caring for a husband with parkinson’s disease and psychotic symptoms. *Social Science & Medicine* 67(4), 583–589 (2008)

Integration of Self-organization and Cooperation Mechanisms to Enhance Service Discovery

Elena del Val, Miguel Rebollo, and Vicente Botti

Universitat Politècnica de València,
Camí de Vera s/n. 46022, València. Spain
{edelval,mrebollo,vbotti}@dsic.upv.es

Abstract. Agents self-organization and cooperation in open societies play an important role in the success of the service discovery process. Self-organization allows agents to deal with dynamic requirements in service demand. Moreover, in distributed environments where service discovery is carried out by agents that only have a partial view of the system, cooperation with neighbors is a key issue in order to locate the required services. However, cooperation is not always present in open agent societies. With this motivation, we present a set of mechanisms that consider self-organization actions and incentives to adapt the structure of the society to the service demand and to promote a cooperative behavior among agents in open societies.

1 Introduction

Service discovery systems are deployed in dynamic environments where their components, features, and tasks do not remain constant. These systems are expected to perform well under many circumstances (i.e., when the number of available agents changes, or when the service demand varies with time). However, the majority of the proposals for service discovery in distributed systems are only focused on the location task and do not take into consideration the inclusion of self-organization mechanisms in order to adapt the social underlying structure to environmental conditions and changes in the requirements [14]. When a global view of the society is not available, these processes should be performed in a decentralized way without the supervision of any central authority. However, these tasks become even more difficult when there are self-interested agents that do not cooperate with others. In that case, if there are no mechanisms to deal with these agents and promote cooperation, the performance of the service discovery process could be seriously compromised [5].

In this paper, we present a combination of self-organization and cooperation mechanisms that agents use in order to maintain the performance of the service discovery process when there are changes in the service demand or when selfish agents appear. The self-organization mechanisms focus on how the relations between agents could be rearranged or how the agent population could be adapted according to the service demand to maintain or improve the performance of the service discovery process. The mechanisms that promote cooperation when there are self-interested agents in the society are based on local structural changes and the use of incentives.

2 Related Work

Search approaches commonly used in decentralized systems where all the entities are considered to be equal and there is an arbitrary topology are based on *blind* or *informed* algorithms. *Blind* algorithms do not consider any information about resource locations and use flooding or random strategies that can overload the system with the traffic generated during the search process [13,18]. *Informed* approaches try to cope with this problem and consider local information to create and guide the search. The information is about their direct neighbors [3,10] or statistics from previous searches and it is stored in local registries [2].

Moreover, the majority of the proposals related to decentralized search or service discovery assume that all the entities that participate in the discovery process are cooperative. However, this not always happen in open societies where there is not a central entity that controls the behavior of the agents. Approaches based on Game Theory have been widely used to explain mechanisms through which cooperation can emerge and be maintained in different scenarios. Depending on the context, mechanisms such as direct reciprocity [11], indirect reciprocity [12], tags [15], or punishment [7] have been used. Many approaches that are based on games assume well-mixed populations where everybody interacts with equal frequency with everybody else. However, real populations are not well-mixed. In real populations, some individuals interact more often than others; therefore, to understand the social behavior of the systems it is important to consider the social structure [6].

The approach that we present in this paper is an informed algorithm that considers both types of local information in order to establish and modify the network structure and to guide the service discovery process. Initially, the structure is created based on the similarity of the resources provided by the agents. However, the environment conditions do not remain constant and in contrast to other proposals that do not consider dynamic environment conditions, in our approach agents consider self-organization actions in order to maintain or improve the performance of the service discovery process when there are changes in the service demand. Unlike other proposals related to self-organization [17], in our proposal, we consider not only changes in the structure of the agents, but also changes in the population of the system. Moreover, we have considered strategies such as incentives and structural changes used in the area of Game Theory in networks to promote cooperation during the service discovery process.

3 Formal Model

Our proposal for agent society is modeled as an undirected network populated by a set of autonomous agents $A = \{i, \dots, n\}$ that establish relationships with other agents $L \subseteq A \times A$, where each link $(i, j) \in L$ indicates the existence of a direct relationship between agent i and agent j based on the semantic similarity of their attributes (i.e., the roles and the services of the agents) [4]. An agent is a social entity that interacts with other agents in the society. It controls its own information about (i) the semantic services it offers $S_i = \{s_i, \dots, s_n\}$, (ii) the organizational roles it plays $R_i = \{r_1, \dots, r_m\}$, and (iii) an internal state st_i , that contains local information used by the self-organization and the

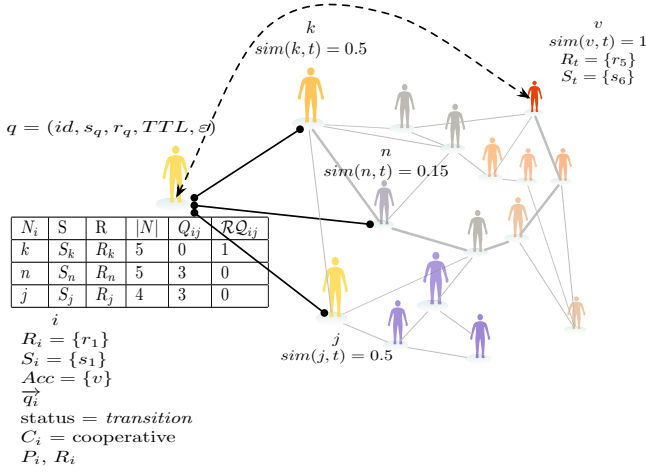


Fig. 1. An example of a decentralized service discovery system where agent i creates a query q that it has to forward to the most promising neighbor (i.e., the closest neighbor to the target).

cooperation promotion mechanisms. The information in the internal state related to the *self-organization* is:

- N_i is the set of direct neighbors agent i has a direct relationship with. For each neighbor $j \in N_i$, agent i has information about: the roles j plays, the services j offers, the degree of connection of j , and the number of times that a query that arrived to the agent i was not forwarded through its neighbor j (Q_{ij});
- Acc a set of acquaintances whose existence agent i is aware as a result of the discovery process but it does not have a direct relationship with;
- $\vec{q}_i = [q_i^{r_1}, q_i^{r_2}, \dots]$ is the local service demand distribution (i.e., the number of queries that the agent receives about services offered by different organizational roles r_1, r_2, \dots);
- the *status* of the agent. The status depends on the significance of the information an agent has. If an agent has an accurate view of the system, it is considered to be in a *stable* situation. When a new agent arrives to the system, or when it has outdated information that introduces noise in its local environment, the agent is considered to be in a *transition* situation.

The information in the internal state of an agent related to the *cooperation* is:

- dc_i represents the degree of cooperation of agent i and ranges in the interval $[0,1]$,
- \mathcal{C}_i represents the behavior of agent i and can take two values: cooperative or not cooperative,
- Q_i is the number of queries that agent i forwarded,
- SQ_i is the number of queries that the agent i forwarded in successful discovery processes,
- RQ_{ij} is the number of queries from agent i that agent j refused to forward,

- \mathcal{P}_i is the number of service requests attended to by agent i ,
- \mathcal{R}_i is the number of service requests sent by agent i .

The decision making process about self-organization actions or actions related to the promotion of cooperation is initiated when agent i generates a query $q = (id, s_q, r_q, TTL, \varepsilon)$, which contains the semantic description of the desired service s_q and the role r_q that the target agent should play (see Figure 1). A *target* agent profile t is created with the service and role specified in the query q . Agent i looks for a neighbor similar to t . If it finds a suitable neighbor, the service discovery process ends. Otherwise, the agent i forwards q to one of its neighbors $j \in N_i$. Specifically, q is forwarded to the agent that has semantic closeness to the *target* agent t and also has a high degree of connection. The selected agent j analyzes based on its payoff and the payoff of its neighbors if it is worthwhile forwarding the query. If j rejects forwarding the query, agent i updates the number of times that agent j rejects its request of forwarding ($\mathcal{R}Q_{ij}$) and based on this, agent i considers breaking its current link with j . If agent j does not cooperate, agent i has to select another neighbor that cooperates in the forwarding process. Once a cooperator neighbor is found, agent i forwards the query to it and updates its information about which of its links have been used. Agent i also updates the number of total queries it received (Q_i), and the number of queries about the role r_q ($\vec{q}_i[r_q]$). When the query reaches a suitable provider agent, all the participants in the service discovery receive a reward. The source agent adds the provider agent found to its set of acquaintances only if it does not already have an acquaintance that plays the role of the provider agent. Finally, the source agent updates its internal state st_i and analyzes the set of self-organization actions that it can carry out.

4 Self-organization Mechanisms

In order to make decisions about self-organization actions agents need to have an accurate local view of the service demand in the system. To evaluate the accuracy of its local view agents analyze its internal state. Initially, an agent is in a *transition* state. An agent in this state does not have reliable and sufficient information to be able to estimate what is the current service demand distribution in the system. In this state, an agent can reorganize its local view of the service demand distribution \vec{q}_i taking into account the number of queries received about the services associated to each role. An agent changes its status from *transition* to *stable* when the correlation degree (ρ_i) between its local data about service demand in the society, \vec{q}_i , and an estimation of the service demand distribution is over a threshold. In our system, the estimation of the service demand distribution follows an exponential distribution. This type of distribution is present in many features of open systems such as Internet [1,8]. Specifically, we assume that the expected service demand distribution is $eDistr(x) = a \cdot e^{x \cdot b}$, where the x parameter represents a role identifier. We estimate the a and b parameters of this distribution using the least squares method and the data from \vec{q}_i . An agent turns back into the *transition* state if it detects a big enough deviation of the correlation degree at any moment. This usually happens when there are frequent changes in the service demand. Once the agent has analyzed its internal state, it is able to make decisions about self-organization actions.

Self-organization of the structural links. Agents are able to reason about whether or not maintain, reinforce or create new structural relations. Agents consider a *decay* metric associated to each link: $decay(Q_{ij}) = 1 - (1/(1 + \cdot e^{-(Q_{ij}-z)/y})$, where y is the slope, z is the displacement constant, Q_{ij} is the number of queries that arrived to agent i and were not forwarded through agent j . Each time agent i forwards a query, it updates the information about the traffic of its links. If the query is forwarded through agent j , the Q_{ij} is updated to 0. Otherwise, the Q_{ij} is increased by increments of 1. With the information provided by the *decay* function, agent i reasons about the benefit of maintaining its current links.

Population self-organization: leaving, remaining, or cloning. The analysis that evaluates whether it is worthwhile for the agent to remain in the system, clone itself, or leave the system takes the following parameters into account: (i)the number of queries received by the agent Q_i ; (ii)the status of the agent; (iii)the structural similarity of the agent SH_i ; (iv)the number of queries forwarded since the last analysis Δq^i ; (v) the degree of correlation ρ_i .

In the context of service discovery, we define the concept of *structural similarity* SH_i as the degree of similarity between the services demanded in the system and the services provided by an agent in the system. This kind of similarity reflects how important an agent is to the system with regard to the current service demand. The structural similarity of an agent with respect the system dynamics is defined by the following function: $SH_i = a \cdot e^{r_i \cdot b}$, where r_i is the role of agent i that maximizes the following function: $r_i = \operatorname{argmax}_{x \in R_i} a \cdot e^{x \cdot b}$, where the a and b parameters are obtained from the local view of the service demand. SH_i ranges in the interval $[0,1]$, where 1 indicates that the services the agent offers are required in the system, and 0 indicates that the services the agent offers are not being demanded in the system. The conditions to select the leaving or cloning actions are shown in Table 1.

Table 1. Parameters and conditions that agents use during the making decision process about self-organization actions. The parameters are: the number of queries received by the agent, the status of the agent, the structural homophily SH_i , the similarity of the neighborhood $\operatorname{SimilarN}(N_i)$, the increase in the number of queries received Δq^i , and correlation value ρ_i .

Action	Num. Queries	Status	SH_i	$\operatorname{SimilarN}(N_i)$	Δq^i	ρ_i
Leave	$1/e^{-(Q_i-d')/y}$	Stable	<	> 0		
Clone	$1/(1 + \cdot e^{-(Q_i-2^{clones})/y})$	Stable	>	< $ N_i $	> 0	> δ

5 Incentives and Social Plasticity

In the proposed model, agents have cooperative or non-cooperative behavior. Cooperating in the service discovery scenario implies that an agent is going to: forward queries, request services, and attend to request about its services. If an agent has non-cooperative behavior, it means that the agent is going to act selfishly by requesting services and offering its services, but it is not going to forward the queries that it receives from its

neighbors. We assume that each action in our model implies a cost and, in order to promote cooperation, the forwarding action has a reward if the search process ends successfully.

Agents in a neighborhood share information about their payoffs. An agent establishes its behavior based on its payoff and the payoff of its neighbors. An agent calculates its payoff as follows: $\mathcal{PO}(st_i) = \mathcal{SQ}_i \cdot sq - \mathcal{Q}_i \cdot q + \mathcal{P}_i \cdot p - \mathcal{R}_i \cdot r$, where $\mathcal{SQ}_i, \mathcal{Q}_i, \mathcal{P}_i, \mathcal{R}_i$ is the information of the internal state (st_i) of an agent; sq is the benefit obtained by the agents that participate by forwarding queries in a service discovery process that ends successfully; q is the cost of forwarding queries; p is the benefit obtained by the agents that provide a service; r is the cost of requesting a service.

The strategy followed by the agents in order to change their behavior is based on imitation [16]. Agents take into account the payoff of their direct neighbors to update their behavior. If an agent has a neighbor that obtains a higher payoff, the agent changes its behavior to the behavior of its neighbor.

When the number of cooperative agents is greater than the number of non-cooperative agents, non-cooperative agents are prone to change their behavior to cooperate since the probability that a query ends successfully is high, and, therefore, cooperation receives a reward if the discovery process ends successfully. However, when the number of non-cooperators is greater than the number of cooperators, cooperative behavior does not always emerge. In order to facilitate the emergence of cooperation in this scenario, in our proposed model, each agent also has the capacity to change its relationships as time passes based on a logistic function that depends on the number of times a neighbor has refused to forward one of its queries. An agent i maintains a counter per each direct neighbor j (\mathcal{RQ}_{ij}) that stores the number of times a neighbor rejected forwarding a query [6]. If a neighbor j decides to change its behavior and forwards queries, the agent updates its counter to 0.

With the combination of the social plasticity and incentives, non-cooperative agents lose connectivity, benefits, and influence in the neighborhood. As a consequence, they decide to change their behavior to the most promising behavior in the neighborhood, which is to cooperate.

6 Experiments

We analyzed the effects of using of self-organization and cooperation mechanisms in the discovery process. The tests were performed on a set of 10 undirected networks with an average degree of connection of 4. The degree of connection distribution follows an exponential distribution. The creation process of the network is described with detail in [4]. The networks were populated by 1,000 agents. The agents played one role and offered one semantic web service associated to this role. Initially, the agents were uniformly distributed over 16 roles, which were defined in an organizational ontology. The set of semantic service descriptions used for the experiments was taken from the OWL-S TC4 test collection¹.

All the agents in the system had the same probability of generating service queries. A query consisted of two features that characterize the required provider agent: the role

¹ <http://www.semwebcentral.org/projects/owls-tc/>

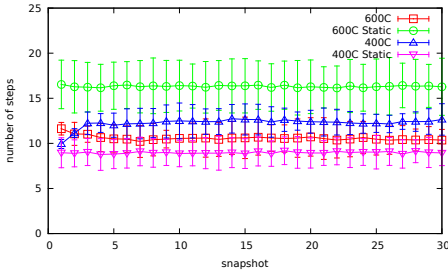
and the service. A query was successfully solved when an agent that offered a similar service (i.e., the degree of semantic match between the semantic service descriptions was over a threshold $\varepsilon = 0.75$) was found before the TTL ($TTL = 100$). Query distribution in the system was modeled as an exponential distribution. In the experiments, we made a snapshot of all the metrics every 10,000 queries in order to see the evolution of the system.

Specifically, the tests focused on a set of metrics that are meaningful for the analysis of the performance of the system and for the effects on the service discovery process when agents incorporate self-organization and cooperation mechanisms [9]. These metrics are: (i) average number of steps required to locate an appropriate agent that solves a query; (ii) % of queries that are solved before the TTL; (iii) communication load improvement (i.e., the system improvement comparing the number of exchanged messages during the service discovery process when adaptation mechanisms are exploited with respect to the number of exchanged messages when the network is not self-organized); (iv) Structural adaptive cost (i.e., the number of structural changes required to adapt the system: number of structural relations between agents that have changed during the service discovery process and number of agents that clone or leave the system during the service discovery process); (iv) evolution of cooperation in the system; (v) number of broken relationships as consequence of social plasticity.

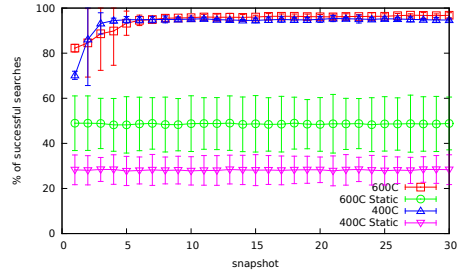
In the experiments, the costs and benefits of the actions were: $q = 0.15$ (cost of forwarding action), $p = 0.5$ (benefit of providing a service), $r = 0.5$ (cost of asking for a service), and $sq = 0.30$ the reward of the forwarding action. The results were evaluated considering two different scenarios. In one scenario the number of initial cooperators in the network was 600. In the other scenario the initial number of cooperators was 400. We compare the results that we obtained using the proposed mechanisms with the results obtained in static networks.

Figure 2a shows the average number of steps in successful searches. In the case of 600 initial cooperators, the introduction of self-organization mechanisms considerably decreased the number of steps required to reach a suitable provider agent if we compare them with the steps required when the network was static and incentives were not considered. In the other scenario, where the initial number of cooperators was 400, the average number of steps increased if we compare it with a static network. This has sense since in static networks with 400 cooperators the only successful queries were those that were solved in the neighborhood of the agent that generated the query. In a dynamic network where mechanisms to adapt to the service demand and to promote cooperation were used, the number of queries solved is higher due to a query that could not be solved by a nearby agent could reach other agents that were far away, therefore the number of steps increased.

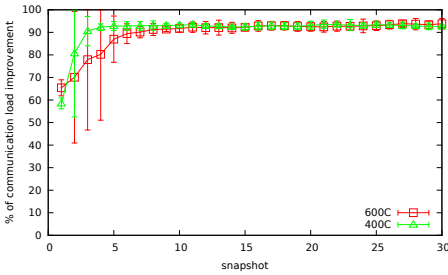
Figure 2b shows the effects of using self-organization and cooperation mechanisms on the success of the service discovery process. In general, the percentage of queries that ended successfully was improved with the inclusion of the mechanisms. This improvement was achieved in the first snapshots where the self-organization and the promotion of cooperation played an important role. The evolution of cooperation can be observed in Figure 2e.



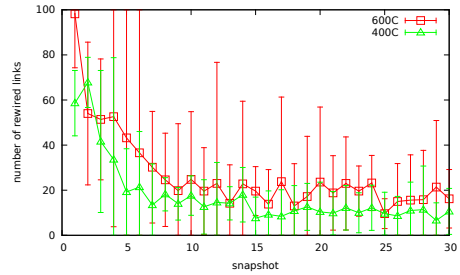
(a) Average path length.



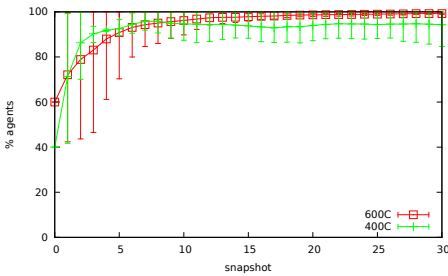
(b) % Successful service discovery processes.



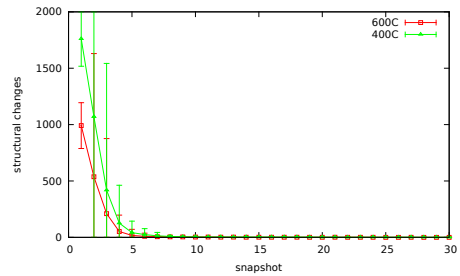
(c) Improvement in the communication load.



(d) Number of changes in structural links.



(e) % of cooperative agents.



(f) Numer of social plasticity actions.

Fig. 2. Effects of the combination of self-organization and cooperation mechanisms in the service discovery process. We considered two scenarios. In one scenario the number of initial cooperators in the network was 600. In the other scenario the initial number of cooperators was 400.

Figure 2c shows the improvement in the communication load. This measures the system improvement comparing the number of exchanged messages during the service discovery process when adaptation mechanisms are exploited with respect to the number of exchanged messages when the system is not self-organized. In general, it can be observed that the number of messages required in a service discovery process was reduced.

Figure 2d shows the number of structural relations that agents change in order to improve the system performance. The results show that the self-organization mechanism

allowed agents to be aware that there was a change in the service demand; therefore, they realized that structural changes were needed to adapt some of their links according to a new service demand. This fact can be observed in the first five snapshots where the number of rewired links was greater than in the following snapshots. In the configuration of 600 initial cooperators, the number of rewired structural relations in the initial snapshots was greater than in the configuration with 400 initial cooperators. This is due to in the network with more cooperator agents, agents had more information about the service demand since more searches ended successfully. As the number of cooperators in the network increased this difference between both initial configurations was reduced.

Figure 2f shows the number of structural changes that were done by the agents in order to isolate non-cooperative agents. In the first 10 snapshots, the number of structural changes was higher than the following snapshots. This fact is because structural changes were used by the agents when the majority of their neighbors were non-cooperators. After the first 10 snapshots, cooperation emerged and in order to maintain the network connected agents only used incentives to promote cooperation.

7 Conclusions

Our proposal addresses the problem of self-organization and cooperation of agents in order to deal with the service discovery when service demand changes or selfish agents appear in open societies. Agents include *self-organization* mechanisms in order to adapt the underlying structure of the agent society to changes in the service demand. Agents replace their relationships with neighbors that are not being used with new structural relations with acquaintances. Agents are also able to estimate whether or not they are playing an important role in the society through the calculation of their structural similarity. With this information, agents decide to remain, leave, or clone themselves in order to adapt the population to the service demand. We also include the use of incentives and social plasticity in order to promote and maintain *cooperation* in the society. Incentives influence the behavior of other agents and promote cooperation. Moreover, social plasticity allows agents to change their structural relations based on the degree of cooperation of their neighbors. We evaluated the integration of the proposed mechanisms through a set of experiments taking into account the effects on the average path length, the percentage of successful searches, the improvement in communications, and the cooperative behavior. The results show that the proposed mechanisms improve the service discovery performance increasing the success, reducing the path length, and increasing the number of cooperators in the agent society.

Acknowledgements. This work is supported by TIN2011-27652-C03-01 and TIN2012-36586-C03-01 projects and FPU AP2008-00601 granted to E. del Val.

References

1. Adamic: Zipf's law and the internet. *Glottometrics* 3, 143–150 (2002)
2. Basters, U., Klusch, M.: RS2D: Fast adaptive search for semantic web services in unstructured P2P networks. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) *ISWC 2006*. LNCS, vol. 4273, pp. 87–100. Springer, Heidelberg (2006)
3. Bianchini, D., Antonellis, V.D., Melchiori, M.: Service-based semantic search in p2p systems, pp. 7–16. *IEEE Computer Society, Los Alamitos* (2009)
4. Del Val, E., Rebollo, M., Botti, V.: Enhancing Decentralized Service Discovery in Open Service-Oriented Multi-Agent Systems. In: *JAAMAS*, pp. 1–30 (2013)
5. Doran, J.E., Franklin, S., Jennings, N.R., Norman, T.J.: On cooperation in multi-agent systems. *The Knowledge Engineering Review* 12, 309–314 (1997)
6. Eguiluz, V.M., et al.: Cooperation and emergence of role differentiation in the dynamics of social networks. *American Journal of Sociology* 110, 977 (2005)
7. Hauert, C., Traulsen, A., Brandt, H., Nowak, M.A., Sigmund, K.: Via Freedom to Coercion: The Emergence of Costly Punishment. *Science* 316(5833), 1905–1907 (2007)
8. Huberman, B.A., Adamic, L.A.: The nature of markets in the www. Technical report (1999)
9. Kaddoum, E., Raibulet, C., Georgé, J.-P., Picard, G., Gleizes, M.-P.: Criteria for the evaluation of self-* systems. In: *Proc. of the SEAMS*, pp. 29–38 (2010)
10. Kontominas, D., Raftopoulou, P., Tryfonopoulos, C., Petrakis, E.G.M.: DS^4 : A distributed social and semantic search system. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) *ECIR 2013*. LNCS, vol. 7814, pp. 832–836. Springer, Heidelberg (2013)
11. Nowak, M.A.: Five Rules for the Evolution of Cooperation. *Science* 314(5805), 1560–1563 (2006)
12. Nowak, M.A., Sigmund, K.: Evolution of indirect reciprocity by image scoring. *Nature* 393(6685), 573–577 (1998)
13. Ouksel, A., Babad, Y., Tesch, T.: Matchmaking software agents in b2b markets. In: *Proc. of the 37th Annual Hawaii International Conference on System Sciences, HICSS 2004* (2004)
14. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: State of the art and research challenges. *Computer* 40, 38–45 (2007)
15. Sigmund, K.: Sympathy and similarity: The evolutionary dynamics of cooperation. *Proceedings of the National Academy of Sciences* 106(21), 8405–8406 (2009)
16. Strang, D., Macy, M.W.: In search of excellence: Fads, success stories, and adaptive emulation. *American Journal of Sociology* 107, 147 (2001)
17. Wang, L.: Sofa: An expert-driven, self-organization peer-to-peer semantic communities for network resource management. *Expert Syst. Appl.* 38(1), 94–105 (2011)
18. Zhong, M.: Popularity-biased random walks for peer-to-peer search under the square-root principle. In: *Proc. of the 5th International Workshop on Peer-to-Peer Systems* (2006)

Agent Participation in Context-Aware Workflows

José M. Fernández-de-Alba, Rubén Fuentes-Fernández, and Juan Pavón

Facultad de Informática de la Universidad Complutense de Madrid
Avda. Complutense, s/n. 28040 Madrid, Spain
{jmfernandezdealba, ruben, jpavon}@fdi.ucm.es

Abstract. Smart environments assist users in the activities taking place in their influence areas. These activities are occasionally part of workflows and have multiple physical or computational participants playing different roles. The system has to monitor the development of the activities, and to take the necessary actions for them and the workflow to reach a certain end. These tasks largely depend on obtaining data from sensors, inferring the proper information from those data, and using actuators consequently. The context-aware paradigm pursues helping to develop these applications. In certain situations, computational participants need to take complex decisions. Agents are a convenient way to describe entities with sophisticated and flexible behaviors that adapt to complex and evolving environments and collaborate to reach certain goals. Most works in this area make use of agents for infrastructure-related or domain-specific tasks, whereas this research proposes patterns to integrate agents on top of an existing context-aware architecture in order to exploit its capabilities to improve functionality. A case study on guiding a user along a path illustrates this approach.

Keywords: software agent, software architecture, context-awareness, workflow management, ambient intelligence, ambient-assisted living.

1 Introduction

Ambient Intelligence (AmI) makes use of different technologies (e.g., location, identity, movement, face or speech recognition), integrated into a myriad of devices. These information sources combined are rich enough to support context-aware systems that adaptively solve high-level tasks minimizing the need of explicit interaction with users. Such tasks frequently involve activity recognition and assistance in business workflows. These workflows may involve multiple actors, including systems and users, which require coordination. The adaptation here implies performing tasks according to the actual setting regarding, for instance, resources and user configurations. A correct evaluation of the setting relies on systems making a proper interpretation of the available data, and using inferred context to fill in the missing information needed by their services.

The previous adaptation requires an infrastructure that solves abstract representations of existing tasks into runtime processes that drive the sensors and

actuators of the smart environment, either to monitor or to perform the needed actions. Also, it needs to integrate the domain logics of different participants, both human and not. A suitable way to design all this information and its processing is using the concept of *software agent*. A software agent is a modeling entity with intentional and social features, that mainly works in terms of information transformations [12]. The integration of agent concept is interesting because there already exist multiple complex tasks studied for this technology that can be applied to AmI problems. These tasks include dialog management, negotiation and collaboration, which may be useful to include into smart environments. Furthermore, the use of agents in an AmI system brings also advantages to agents: they gain access to a generic and shared source of information (i.e., the *context*), so less interactions among them are required to obtain certain information.

Although there are solutions for smart environments with context-aware agents, they present some limitations. Many of them are focused on low abstraction levels or very domain-specific tasks [11]. Therefore, they lack of general patterns to organize agents in order to work with abstract context representations, including abstract workflow definitions. Moreover, the complexity of the considered activities is usually quite limited in terms of types of actions and number of participants involved. This produces some uncertainty regarding the scalability of the proposals in this aspect.

To address this problem, this paper presents an architecture to integrate software agents on top of an existing framework for smart environments. The design of these agents may use common agent design concepts, as extracted from the INGENIAS agent-oriented methodology [7]. This work extends FAERIE (Framework for AmI: Extensible Resources for Intelligent Environments) [3], which supports the development of smart environments with a distributed management of context and workflows. It proposes splitting the context representation in abstraction layers. Each layer considers certain information and its processing, both horizontal (i.e., in the same abstraction layer) and vertical (i.e., between successive abstraction layers). FAERIE establishes how to coordinate the components of these layers in order to process information, which facilitates context reasoning. The workflow management, i.e., detection, tracking and execution of workflows, is built upon the previous functionality. Workflows are represented as activity diagrams that work using abstract expressions as data. The context-aware applications are constructed over the previous functionality.

A case study of a system that guides a user following a path and making certain actions illustrates the use of these patterns. The system has information about the building map and an activity diagram that describes the guiding process. It updates the workflow status to track the user and to deduce the completion degree of the workflow. The infrastructure coordinates the available sensors and actuators to perform the actions described in the workflow. A software agent is included in the workflow as the actor responsible for the guiding activity, which makes use of dialog management in order to maintain an interaction with the user.

The rest of the paper is organized as follows. Section 3 presents the pattern structure for the integration of software agents. Section 4 illustrates its use with the guiding case study and shows a complete execution of the workflow. These results are used in Section 2 to discuss alternative approaches to deal with context and workflow management in context-aware systems and their tradeoffs. Finally, Section 5 presents some conclusions and future work on the approach.

2 Related Work

With respect to context-aware workflows management, most solutions clearly separate the control of the workflow execution and the management of context information. Ranganathan et al. [10], uFlow [8] and CAWE [4] are examples of this. The first two propose wrapping the context management system and using it to check conditions in a workflow execution environment, while the third wraps the workflow management as another context provider/consumer into a context-aware architecture. The Ranganathan et al.' approach uses the context-aware component in order to choose a suitable workflow definition, and then proceeds to its execution. However, it does not consider the change of the circumstances after the definition has been chosen. On the contrary, the other two approaches work with an abstract workflow definition, which actions are instantiated at runtime depending on current context conditions. The approach chosen in FAERIE is similar to uFlow, but it supports the use of any workflow definition language, as the correspondent engine is wrapped with modules that adapt the inputs and outputs as required. This is not the case of uFlow and CAWE, which describe some definition languages of their own, and of Ranganathan et al., which propose using BPEL, a standard language for the definition of business processes. The advantage of the first two alternatives over the third is that they include context-dependent concepts, which allows defining the workflows conditions and actions in terms of the context information. However, the third alternative uses a well-known and established language, which facilitates its reuse.

The use of the agent paradigm is also widespread in AmI literature [11]. A notable example is Aiello et al.' project [1], which describes a framework to develop Wireless Sensor Networks (WSN) using mobile agents. It does not provide support for defining and monitoring workflows. Instead, it considers the detection of activities by means of body sensors. The definition of these activities is mainly achieved through assisted automatic learning, as there is not a explicit definition of them. This offers a great flexibility to monitor activities, and to change their definition in runtime. The drawback is the limited complexity of the learned activities regarding types of actions and number of participants. Other example is CAKE [5], in which agents organize themselves using abstract workflow definitions, depending on the situation obtained from a case-based reasoner. However, the architecture does not offer specific support to facilitate information interpretation in order to identify the suitable case. Instead, it relies in the implementation of the agents to determine the situation to requesting the case.

Next section explains how the FAERIE framework [3] is extended with the integration of software agents, as computational entities with an autonomous behavior and capable of communication with other agents, forming Multi-Agent Systems (MAS) [12]. When it comes to workflows, agents may assume the role of certain participants, receiving pieces of information and performing tasks, thus making the workflow progress to a certain end. The main difference with the previous approaches is that the agents participate in workflows in an abstract manner, taking advantage of the features provided by the context management framework to monitor abstract conditions. In addition, the management of its internal information is defined in order to use this features in a transparent way.

3 Architecture and Integration

The architecture of FAERIE is inspired on distributed blackboard models [6]. It conceives a system as a set of interconnected *environments*, each one with its own blackboard (i.e., the *context container*) and a set of components working on it (i.e., the *context observers*). This works the following way: when certain context observer wants to perform a task that needs some information from the context, it makes a request to the context container and waits its response. Upon this request, the context container publishes the requirement of information and notifies the other context observers. If any of these context observers provide the required information directly, then, the request is responded. If not, some context observers may need to make successive requests for information, including requests to remote environments to ultimately provide the required information. The overall system is a federation of *environments* that supports component collaboration through shared information.

The functionality of the FAERIE architecture is divided in layers. The *Component Infrastructure* layer integrates services from standard component-based frameworks: life cycle of components, and dynamic discovery and binding of abstract services. The *Context Management* layer offers mechanisms to facilitate context accessing and manipulation. The main services provided by this layer are the *request* and *subscribe* mechanisms. The framework handles the creation of information flows among the different components that are responsible to interpret and transform the knowledge to different levels of abstraction, as explained at the beginning. An additional mechanisms this layer provides is a way to declare behavior changes depending on context conditions, by making use of the *subscribe* service to check these conditions [2]. Finally, the *Workflow Management* layer includes the patterns to manage the status of an activity represented as a workflow, which will be explained below. The final and upper layer, which does not belong to FAERIE, is where context-aware applications are built using the services of the previous layers.

Following the pattern described by the FAERIE architecture, the workflow management works this way: given a workflow definition (i.e., a set of interconnected activities described in a certain language) and a workflow engine capable of interpreting it and creating the corresponding signals, the pattern creates

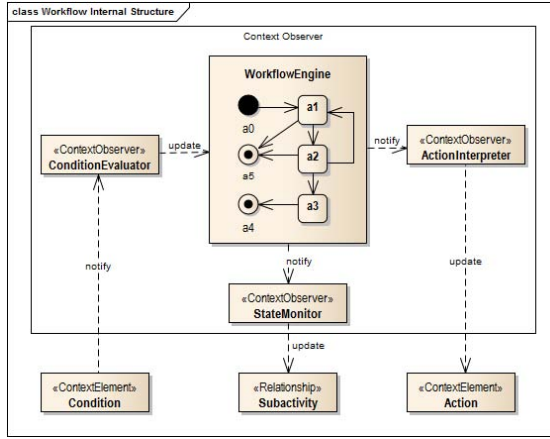


Fig. 1. Workflow pattern structure

four modules. These are *Condition Evaluator*, *Workflow Engine*, *Action Interpreter* and *State Monitor*. These are in charge of respectively: interpreting the abstract conditions on the context necessary to update the workflow; updating the runtime representation of the workflow and checking the next required actions; requesting the required actions to the context management; and providing the current state of the workflow as a context information. Figure 1 shows the instantiation of this pattern for a concrete workflow, being a1, a2, etc., its corresponding activities.

This is the point where the agents are integrated. Different implementations for software agents may be used, as long as they provide the *context observer* interface. This makes them able to observe and change context information, and define different behaviors depending on context conditions. This way, agents are benefited from the capabilities of the framework to deal with changing resources, and therefore their design may be done in a more abstract way.

Also, as in many agent theories, individual agents possess their own “mental state”. This mental state acts for the agent as a private context container where it manages its own information. The architecture integrates this mental state with the management of the rest of the context in the *context containers*, as shown in Fig. 2. This *mental state container* acts as the previously described *context container*. Each time an agent creates an entity to update the mental representation of its own and environment states, it is able to specify whether the entity is created publicly or privately. When it does this, the agent will see if itself is capable of resolving the request, and if not, the request will be forwarded to the *context container* of its node. This mechanism works the same way as when the local context container ask a remote environment for information. This allows preventing the node *context container* from storing information that is not interesting or public for other components of the node. On the other hand, the existence of a *context container* accessible by different agents reduces

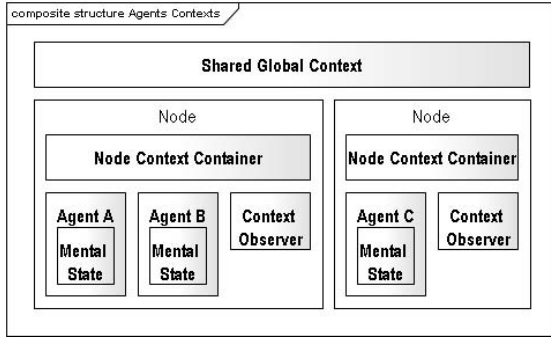


Fig. 2. Structure of the context containers in a distributed system

the exchange of messages to “inform” other agents, since they have access to most of the basic information of the current *environment*. The upper layer in Fig. 2 represents the abstract global container. It is the part of the context from different nodes that they share with other nodes of the distributed system. This container does not exist physically, as it is the result of the actions of the information sharing mechanisms among nodes.

The design and mental state of agents are represented using common concepts of the agent paradigm. Figure 3 shows an example from the case study following the INGENIAS notation [7]. The definition of the *GuideAgent* considers the *goals* it pursues, and the *tasks* it can perform and the *workflows* where it participates to achieve those goals. Tasks produce and consume pieces of information (e.g., *events* and *frame facts*), and use resources (e.g., *resources* and *applications*). Different agent-oriented infrastructures support working with these elements. In the case of INGENIAS [7], the INGENIAS Development Kit (IDK) is used for modeling and code generation, and the INGENIAS Agent Framework (IAF) provides the basic libraries for the agent code and debugging tools.

Exploiting the abstraction level and infrastructure provided by the agent paradigm facilitate the development of complex AmI scenarios, which may involve multi-agent interactions, automated learning, planning, and decision making. The execution of simple tasks will be handled by basic FAERIE *context observers*, while the most complex tasks will be defined by means of MAS.

4 Case Study: Teacher Finding

To illustrate the previous patterns, this case study considers a system that guides students to the room where a tuition hour will take place. This allows the school to arrange dynamically rooms for tuition according to the actual number of assistants and the required resources, or the room where the teacher is (e.g., a laboratory or an ongoing tuition class). The application is responsible to find out the student and teacher locations, and to guide the student through the building to the target place. The system uses spoken information for guidance, which is

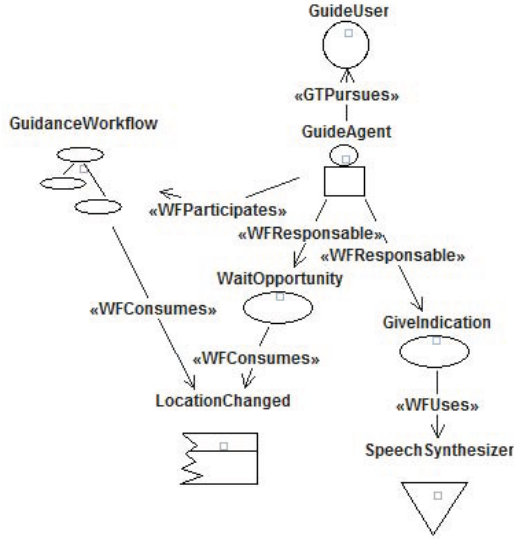


Fig. 3. Model of goal and tasks for an example agent

interactive and opportunistic. This means that the indications are activated only if it is considered necessary by the system, or asked by the user via voice. This may produce a dialog between the system and the user. The tracking uses the user's location to monitor that certain activities has been completed.

The responsible of handling the guidance task is defined as a software agent (i.e., the *GuideAgent*). The kind of proposed autonomous and reflexive behavior is easier modeled and developed in terms of agent theory, i.e., using goals, protocols, interactions. The work of this agent takes as basis the formal definition of the workflow and the infrastructure provided by FAERIE. Using this, the agent is able to read the current state of the workflow, and the necessary information from the context in order to establish a dialog with the user. This dialog is a complex task that needs to take into consideration specific intelligence, such as dialog management, and natural language understanding. Its implementation using agents is commonly studied in the literature [9].

Figure 4 represents the runtime behavior of the *GuidanceWorkflow* component and its collaborations. It includes the initialization and evaluation of the context elements necessary to update the progress level of the workflow. The involved steps are:

1. The *GuidanceWorkflow* starts and updates the context to indicate that its current subactivity is "findTeacher".
2. The *Context Container* publishes the change.
3. The *GuideAgent* discovers that there is a subactivity taking place. This causes the agent to create a goal to assist the user in finding the teacher, as described previously.

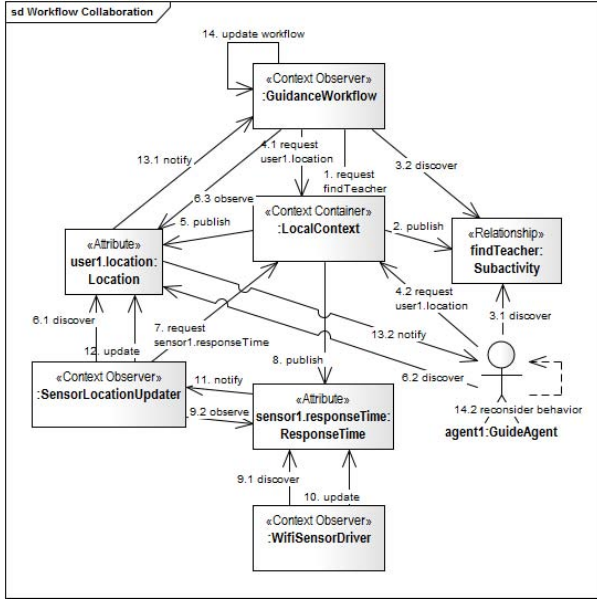


Fig. 4. Collaboration diagram for the workflow pattern in the guidance system

4. The *GuidanceWorkflow* component requests the *user1.location* element, since it needs that information to calculate if the “findTeacher” subactivity has finished. The *GuideAgent* also requests the *user1.location* element. It needs this information to complete its objective.
5. The *Context* publishes the request.
6. The *GuidanceWorkflow* and the *GuideAgent* start to observe the element, and a *SensorLocationUpdater* discovers it.
7. A *SensorLocationUpdater* triangulates the user position from the response time of wifi sensors to the signal of the user’s smartphone. Thus, it needs to find out a *sensor1.responseTime* to calculate the position, and it requests it to the *Context*.
8. and subsequent. The framework coordinates the different components (i.e., the *WifiSensorDriver*, *SensorLocationUpdater* and *GuidanceWorkflow*) to update the context according to the information retrieved by the sensors. The same is done with any other information that the agent may need from the environment, in this case this would be the user’s voice input, and his language and communication preferences.

In the proposed process, once that the bindings have been established, the context representation is changed dynamically by the system components to reflect the progress of the situation. Under this conditions, the agent develops its dialog management in complete abstraction about the actual coordination or information fusion mechanisms that are being used to provide that information. The lower layers hide their specific details to the upper layers. For instance, as the user

walks through the room, the sensors produce a lot of information. The *SensorLocationUpdater* processes this information, but it only updates the *user1.location* element when the sensor context reflect a change of position. The same is done at the activity level, as the workflow component only determines the end of an activity and starts a new one when the position reaches a given place.

As the *GuideAgent* is free of the details for obtaining information, it is possible to focus its development on more complex tasks. These task would include collaboration, negotiation, and abstract reasoning, fields largely studied in the agent literature.

5 Conclusions

This paper has introduced a way to integrate software agents into a generic architecture for context-awareness and activity management in AmI systems. Its definition includes patterns to develop components that create, manage and use elements of information in the context. Based on these components, it also proposes patterns to develop subsystems that are able to track context conditions and to perform different actions according to activity diagrams.

Considering the integration of MAS and workflow management facilities provides advantages to both types of systems by increasing the abstraction level of behavior definition. The specification of software agents use concepts such as goals, protocols and interactions, which are useful to describe declaratively workflows and their relation with actors' purposes. This facilitates considering in AmI systems the definition of more complex workflows and automated reasoning on them. Also, the existence of a common "mental state" for agent organizations using the AmI context reduces the need of *inform* interactions among agents. The main advantage over previous approaches is that it uses agents to participate in workflows in a transparent and generic way, i.e., the agents do not need to know neither how the conditions are resolved nor the concrete workflow in which they are participating.

Open issues of this work include completing the FAERIE architecture and infrastructure for AmI applications with MAS. The architecture will include a *Security Management* pattern, in order to guarantee the correct access to sensitive information in context containers. Also, a future case study, would consider several students trying to meet with the teacher at the same time. This conflict would be resolved through negotiation among the different student agents and a teacher agent, in order to collectively arrange the corresponding meetings. This collaboration will use interaction protocols, and may be benefited from the shared contexts described in the architecture section.

Acknowledgments. This work has been done in the context of the project "Social Ambient Assisting Living - Methods (SociAAL)" (TIN2011-28335-C02-01) supported by the Spanish Ministry for Economy and Competitiveness. Also, we acknowledge support from the "Red Científico-Tecnológica en Ciencias de los Servicios" (TIN2011-15497-E) and the "Programa de Creación y Consolidación de Grupos de Investigación" (UCM-BSCH GR35/10-A).

References

1. Aiello, F., Fortino, G., Gravina, R., Guerrieri, A.: A Java-Based Agent Platform for Programming Wireless Sensor Networks. *Computer Journal* 54(3), 439–454 (2011)
2. Fernández-de-Alba, J.M., Campillo, P., Fuentes-Fernández, R., Pavón, J.: Opportunistic Sensor Interpretation in a Virtual Smart Environment. In: Yin, H., Costa, J.A.F., Barreto, G. (eds.) *IDEAL 2012*. LNCS, vol. 7435, pp. 109–116. Springer, Heidelberg (2012)
3. Fernández-de-Alba, J.M., Fuentes-Fernández, R., Pavón, J.: Dynamic Workflow Management for Context-Aware Systems. In: Novais, P., Hallenborg, K., Tapia, D.I., Rodríguez, J.M.C. (eds.) *Ambient Intelligence - Software and Applications*. AISC, vol. 153, pp. 181–188. Springer, Heidelberg (2012)
4. Ardissono, L., Furnari, R., Goy, A., Petrone, G., Segnan, M.: Context-Aware Workflow Management. In: Baresi, L., Fraternali, P., Houben, G.-J. (eds.) *ICWE 2007*. LNCS, vol. 4607, pp. 47–52. Springer, Heidelberg (2007)
5. Bergmann, R.: Ambient intelligence for decision making in fire service organizations. In: Schiele, B., Dey, A.K., Gellersen, H., de Ruyter, B., Tscheligi, M., Wichert, R., Aarts, E., Buchmann, A.P. (eds.) *AmI 2007*. LNCS, vol. 4794, pp. 73–90. Springer, Heidelberg (2007)
6. Corkill, D.: Blackboard systems. *AI Expert* 6(9), 40–47 (1991)
7. Gómez-Sanz, J.J., Fernández, C.R., Arroyo, J.: Model Driven Development and Simulations with the INGENIAS Agent Framework. *Simulation Modelling Practice and Theory* 18(10), 1468–1482 (2010)
8. Han, J., Cho, Y., Kim, E., Choi, J.-Y.: A Ubiquitous Workflow Service Framework. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) *ICCSA 2006*. LNCS, vol. 3983, pp. 30–39. Springer, Heidelberg (2006)
9. Kopp, S., Gesellensetter, L., Krämer, N.C., Wachsmuth, I.: A conversational agent as museum guide – design and evaluation of a real-world application. In: Panayiotopoulos, T., Gratch, J., Aylett, R.S., Ballin, D., Olivier, P., Rist, T. (eds.) *IVA 2005*. LNCS (LNAI), vol. 3661, pp. 329–343. Springer, Heidelberg (2005)
10. Ranganathan, A., McFaddin, S.: Using Workflows to Coordinate Web Services in Pervasive Computing Environments. In: 2004 IEEE International Conference on Web Services (ICWS 2004), pp. 288–295. IEEE Computer Society, Washington, DC (2004)
11. Venturini, V., Carbó, J., Molina, J.M.: Methodological Design and Comparative Evaluation of a MAS Providing AmI. *Expert Systems with Applications* 39(12), 10656–10673 (2012)
12. Weiss, G.: *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press (1999)

PHAT: Physical Human Activity Tester

Pablo Campillo-Sanchez¹, Jorge J. Gómez-Sanz¹, and Juan A. Botía²

¹ Facultad de Informática,
Universidad Complutense de Madrid, 28040 Madrid, Spain
pabcampi@ucm.es,
jggomez@fdi.ucm.es
<http://grasia.fdi.ucm.es>

² Facultad de Informática,
Universidad de Murcia, 30100 Murcia, Spain
juanbot@um.es

Abstract. This paper introduces PHAT, a 3D environment for facilitating the development of AAL services by providing a virtual living lab. PHAT's purpose is reduce development costs of these services. It is based on an existing 3D game engine, jMonkeyEngine, to provide realistic representations of scenarios including visual details and physics laws to make the simulations more credible. The 3D engine is combined with MASON, a Multi-Agent System Based Simulation tool. MASON is used to create repeatable experiments and facilitate the testing in controlled situations. The paper shows a proof of concept of the MASON-jMonkeyEngine where some unexpected behaviours are modelled thanks to the properties of the 3D engine.

Keywords: Modelling, Parkinson's Disease, Simulation, Mixed Reality, Multi-Agent Based Simulation.

1 Introduction

World's population of 60 years age and older has doubled since the eighties. Furthermore, the World Health Organization forecasts that they will reach 2 billion by 2050¹. In this scenario, the incidence of age related diseases such as Parkinson's disease (PD) will clearly raise. PD is a neurodegenerative disease that affects a part of the brain which is involved with the control of movement and generates a progressive difficulty for carrying out an autonomous everyday life. The quality of living of Parkinson's patients can be seriously affected.

For them, and people with similar difficulties, Ambient Assisted Living (AAL) was devised. In the case of PD, specific AAL solutions are not usual, though. This work takes part of the SociAAL² (Social Ambient Assisted Living) project, which aims at the affordable creation of customised software for people with concrete disabilities due to PD.

¹ http://www.who.int/features/factfiles/ageing/ageing_facts/en/index.html

² SociAAL project website: <http://grasia.fdi.ucm.es/sociaal/>

It is our working hypothesis that generic AAL systems are not completely satisfactory for Parkinson's patients. There are some specific problems with PD that deserve some additional consideration, such as the akinesia situations where the patient just cannot move, even from an upright position. However, the development of solutions for AAL is expensive partially due to the cost of field trials using living labs or volunteers. This slows down the development and, consequently, increases the personnel costs, and the time to market of any commercial, and affordable, solution.

A way to test in advance AAL concepts is to use simulators, e.g. UbikSim³. They intend to reproduce an environment where hardware can be plugged into the simulation directly, and their control software can receive inputs from the different virtually installed devices [1]. Nevertheless, UbikSim and, other similar simulators, seems unreal to the external observer, perhaps due to the lack of realism regarding physical and kinematic aspects. Situations where the caregiver does not hear the caretaker because of the house walls, or accidents where the caregiver becomes the one to receive aid, cannot be represented in UbikSim as it is. However, bringing those possible outcomes to the simulation, would surely advance problems before end-users find them.

Another working hypothesis is that work done in 3D games can help building more effective simulators for AAL. There is a shared goal between our intended AAL simulator and current 3D games. It is easy to appreciate most 3D games look for bringing realism to the gamer, too. Consequently, there has been a huge effort to develop rendering engines capable of drawing convincing 3D scenarios and also physics engines telling how the 3D meshes are expected to interact. This makes possible pushing an object or tripping over some obstacle. Henceforth, using as starting point jMonkeyEngine⁴ (jME), a 3D game engine with its development kit, we introduce a preliminary version of the Physical Human Activity Tester (PHAT).

PHAT is a evolution of the ideas behind UbikSim, developed from scratch to create scenarios where simulated humans recreate activities of daily living and with the capability of showing how the AAL system under development is going to be used. PHAT's purpose is reduce development costs of AAL services. For this, PHAT is focusing on two task: (1) improve requirements elicitation, the simulator would be able to reproduce requirements in a real-time 3D animation in order to social workers can check if what is displayed is what they want, and (2) requirements validation, the simulator would allow an application to check both if the interaction behaviour of the characters with the application conforms to the expected and if the application can properly treat variations of the scenarios collected. Working at these both levels would be expected to reduce (1) the waiting times in development due to misunderstandings and (2) the use of living lab, thanks to detection of early-trivial mistakes.

PHAT is not limited to just reusing 3D engines, but it also integrates with MASON[2] for the sake of achieving a more controlled behaviour. MASON serves

³ UbikSim website: <http://ubiksim.sourceforge.net/>

⁴ jME website: <http://www.jmonkeyengine.org>

to organise the behaviour of the actors by assigning them a MASON agent. MASON simulation capabilities are combined too with the 3D engine in order to create repeatable experiment sequences.

The paper is organised as follows. First, we introduce some related works in section 2. Then, section 3 introduces PHAT by accounting which features of 3D game engines are relevant, see section 3.1, what is the relevance of discrete event simulation in this problem, section 3.2, and then how both concepts can be used to test AAL applications once a scenario capturing system requirements is prepared, see section 3.3. A case study to validate the jME-MASON integration is introduced in section 4. Finally, there are conclusions in section 5.

2 Related Works

There are not much works that use simulation to test AAL applications or to requirements elicitation. Naranjo et al. [3] propose a modelling framework that facilitates and streamlines the process of creation, design, construction and deployment of technological solutions in the context of AAL assuring that they are accessible and usable for elderly people. They define two environments (to be used in a *user centered design methodology*): (1) an *authoring environment* that allows the definition of the user, environment and service models, and (2) a simulation environment used to implement actual *Virtual Reality* (VR) scenarios of AAL that will be used to verify interactions designs and validate the accessibility of AAL products by means of immersing the users in 3D virtual spaces. Sala et al. [4] extend the idea of the previous work. Both works present the results of VAALID project in developing this approach of creating tools for design and simulation of AAL Solutions using VR and Mixed Reality, supporting the early involvements of beneficiaries in the process. However, the project is centred in VR using InstantReality⁵ where the definition of the scenarios is closed and deterministic due to, in part, lack of a physics engine. Besides, their models are concerned with AAL applications and user interaction with the applications but not with the daily activities and her behaviour in the physical space. Therefore, this approach does not allow automatic tests because a user is always necessary.

Campillo-Sanchez et al. [5] extends UbikSim framework in order to be able to test smart phone applications using virtual environment. Compiling the application with their library, it could be installed on either a smart phone or an emulator. The library allows the application to be connected to the virtual world transparently. The test process is carried out automatically since user behaviour is modelled using an hierarchical automaton. However, the virtual environment performed by UbikSim has some deficiencies due to lack of physics engine, light and animation system. So, simulating an accelerometer or a light sensor of the smart phone is a hard task using UbikSim.

Shoulson et al. [6] present ADAPT⁶ which is a flexible platform for designing and authoring functional, purposeful human characters in a rich virtual

⁵ <http://www.instantreality.org/>

⁶ ADAPT website: <http://cg.cis.upenn.edu/hms/research/ADAPT/>

environment. Their framework incorporates interesting facilities such as character animation, navigation, and behaviour with modular interchangeable components to produce narrative scenes. Also, they present a behaviour framework that allows a user to fully leverage the above capabilities using a centralised and event-driven model. This work could be a good starting point for developing PHAT. However, it is developed using the popular Unity 3D game engine which is neither open source nor free.

3 PHAT - Physical Human Activity Tester

PHAT intends to work as an inexpensive virtual living lab. Developers will use it to carry out requisites elicitation or to test AAL services. Hence, it is important that the environment convinces a human observer at the same time it remains developer friendly. For us, it means incorporating five features: (1) *Repeatable experiments*. It is not convenient that the environment is non-deterministic every-time. To some extent, experiments with PHAT ought to be repeatable. Hence, the developer ought to be able to have determinism when required; (2) *Convincing*. It should behave as one would expect from a real world scenario. This living lab should reproduce the activities of daily living of the actors which have been identified in the real world. It also capture those behaviours and circumstances the developer did not foresee. (3) *Configurable*. It should be easy to define concrete living labs, determining specific features of the actors and the environment, as well as their expected behaviour and undesired consequences. (4) *Visual and Observable*. The actual living lab ought to be visualised in a friendly way, but also to associate monitors to observe concrete events. (5) *Pluggable*. The environment should be capable of incorporating simulated or real AAL systems. Being a living lab, one wants to run an AAL system inside.

We started from a previous experience of UbikSim which satisfied previous requirements to a great extent, except the *convincing* one. The environment behaviour was hard-wired, so it did not have to match the real world's one. Physics in UbikSim did not exist, and actors were very simple animations.

In this work, we start from a focus on tools for reproducing real live conditions, and one of the most advanced lines in this direction are game engines. Game Engines are a good starting point for developing a realistic dynamic environment such as Section 3.1 argues. Game engines do not have to lead to repeatable experiments platforms, though. Hence, we considered reusing Mason, a discrete event simulation engine based in Multi-Agent Systems (MAS) to provide the repeatability feature. Section 3.2 presents a design about how to combine PHAT with a discrete event simulation of agents. The logic of the simulation is performed by the agents which take advantage of game engine capabilities.

3.1 Game Engine

A game engine is a complex piece of software. Due to this complexity, there is a great interest in promoting software reuse and finding the right way to

organise the necessary elements to create a game. This has led to game engines written for a specific game but general enough to be used for a family of similar games [7]. A Game Engine is a collection of modules of simulation code that do not directly specify the game’s behaviour (game logic) or game’s environment (level data). Game engines are built in layers and are large-scale systems. They generally consists of a tool suite and a runtime component. Figure 1 shows a typical runtime game engine architecture.

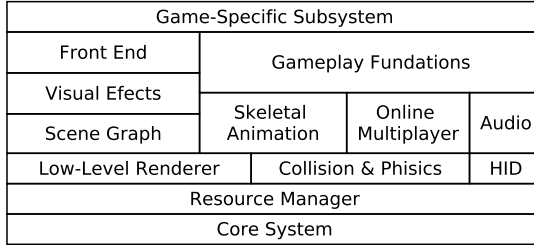


Fig. 1. A simplification of a typical runtime game engine architecture figure in [8]

Looking at Figure 1, and pursuing to create a virtual living lab, we found relevant the features provided by the following components:

- *Resource Manager* provides a unified interface for accessing any and all type of game assets. Assets are the resources we can use to model the different parts of the living lab, including the house and the actors.
- *Rendering Engine* is one of the largest and most complex components of any game engine. This engine includes a low-level renderer, a scene graph, visual effects (such as particle systems and light mapping) and front end (e.g. an in-game graphical user interface). This rendering makes possible a configurable human friendly visualization of the environment. It also permits to determine which parts are visible from the perspective of individual actors (either because there are obstacles or low lights) and, that way, simulate better the limitations of human vision in the real world.
- *Collision and Physics* are usually quite tightly coupled and offer a realistic dynamics simulation creating a space where elements behave according physics laws. Physics are one of the parts that interest us the most, since they make the living lab more credible. It permits the observer to approve the results of some actions.
- *Skeletal Animation* permits a detailed 3D character mesh to be posed by an animator using a relatively simple system of bones. Animations are relevant to reproduce problems in actors, like the gait problem in Parkinson’s patients.
- *Human Interface Devices (HID)* allow game engine to process inputs from the player. Inputs in our case are limited, but we would expect the human observer to trigger some specific events to watch what the actors ought to do.

- *Audio* is just as important as graphics in any game engine. Audio immerses the player in the virtual world. In the living lab, proper audio reproduction includes fade-in and fade-out, perhaps because of physical obstacles, which permits to simulate a person does not hear another.
- *Online Multiplayer* module permits multiple human players to play within a single virtual world. In our case, it may be used to let humans play the roles of the actors in the scenario to quickly test one situation which is not coded yet.
- *Gameplay Foundation System* is the layer to bridge the gap between the game-play code (the action that takes place in the game) and the low-level engine systems discussed thus far. We include here the integration with the discrete event simulation engine which will provide the execution repeatability.
- *Game-Specific Subsystem* contains the game specific code. In our living lab, it would contain the control mechanisms to code the actors behaviours, as well as the AAL software to be tested within the game.

These features can be found in many game engines. Some are commercial, but others are free software. As game engine, this work uses jME, which is developed in Java under a BSD license. jME includes the lightweight OpenGL Java library (LWJGL), under BSD license, for rendering and jBullet, under ZLib license, as physics engine. jME is very well documented and supported. It also regards the above-mentioned features. Nevertheless, working with jME is like creating a game, with all the implicit challenges a game development has. A person interested in testing AAL applications ought to be unaware of such complexities, and this is where PHAT does its part. It intends to take advantage of those features in a developer friendly way.

3.2 Discrete Event Simulation

A simulation is the imitation of the operation of a real-world process or system over time. Besides, a discrete-event system simulation is the modelling of systems in which the state variable changes only at a discrete set of points in time [9]. Designing a simulation is not trivial and requires, among others, to define precisely the purpose of the simulation. In our case, it is reproducing an intended behaviour as expressed in the requirements. Basically, in the simulation, a sequence of events is found and the participating actors do perform some pre-arranged actions. The characterisation of these actors as well as the description of their actions is very relevant to our problem and very related to the research in Multi-Agent Based Simulation (MABS).

While it is common to talk about *intelligent bots* in game developments, in our case we want to have too *stupid bots* that repeat sequences of actions. While the first kind may capture the “unexpected actions” one may desire, the second are more suitable for thorough experimentation. The separation between both is a matter of how bots are actually programmed. In any case, we would like

some control over the different actors and how they interact. It should be clarified that one actor may be a body or one object. Some interactions are not controlled explicitly, like physical interaction due to collisions, because it is the purpose of this effort to add realism. So, we let collisions happen and we decide if there are additional consequences to those pointed out by the physical engine, like the act of pushing a button.

It is hard to balance the different sources of control when part is retained by the game engine. Our solution has been through the extension of the game cycle. A game engine do have a game cycle, just as a simulation engine has a simulation cycle. By triggering a simulation cycle whenever the game cycle is run, it can be obtained a rather straightforward integration. Once linked the discrete event simulation and the game, it is still necessary to define the simulation objects. Since we need to define the interplay of different actors which are capable of acting over the environment, a MABS approach was the obvious choice. From the existing solutions, we opted for MASON⁷ due to the positive experience with UbikSim. MASON is a fast discrete-event multiagent simulation library core in Java.

MASON enables repeatable simulations. MABS means, on the other hand, that it is possible to use the agent metaphor to model the dynamic entities taking part on the simulation. With MASON, the entity agent is used as the main modelling element.

The drawback of MASON is that many of its features, like their own 3D space representation, are hard to fit when dealing directly with something as complex as jME. Nevertheless, MASON can provide a simplified, non 3D, vision of the situation. Also, in situations with a high number of agents, let us remind that agents in MASON can represent humans or objects in jME, it could be observable some emergent behaviour. A system is said to have emergence when it shows dynamic behaviours that were not specified at design time. Precisely, emergence is another interesting means to reproduce the unexpected, i.e. to generate behaviour that was not designed beforehand. In summary, the following three reasons make MASON very appealing within PHAT: (1) agents and multi-agent systems as a modelling metaphor, (2) repeatable experiments and (3) emergence.

3.3 Testing AAL Services

PHAT intends to serve as virtual living lab. Henceforth, it should permit the incorporation of AAL services to this virtual reality. An AAL service will be typically made of a network of sensors plus one or many processor units. PHAT must offer interfaces in order to implements these sensors (e.g. a thermometer, accelerometer or light sensor) and actuators (e.g. a switch) and real AAL services can be tested using a virtual world.

Smart phones are devices equipped with a broad range of sensors and wireless connectivity. These capabilities, and their ever increasing popularity, make these

⁷ MASON website: <http://http://cs.gmu.edu/~eclab/projects/mason/>

devices a fitting choice for installing AAL services on them. So, we intend to focus first on these devices in PHAT. Being an AAL developer using smart phones as target platform, we expect PHAT to provide: (1) Sensor simulation. Every sensor of smart phone must be simulated in order to the service perceives its context in the virtual world. A game engine helps to overcome this challenge. PHAT should publish mechanisms to allow devices create and connect to them. (2) Actuator simulation. Actuators in a smart phone are speaker and flash light, and PHAT should publish mechanisms like sensors also. (3) Simulated smart phones. They are a composition of sensor and actuators. Also, an AAL service could have a GUI, so the agent need to interact with the GUI pressing buttons and interpreting outputs in the screen. PHAT need to model this feature in order to agents can interact with the smart phone application. If the service uses a voice interface, this is related with the first feature, and could be solved with a simulated microphone sensor (using audio engine) and a text-to-speech library.

Once the AAL service is integrated, the testing itself ought to proceed. A test, in the classical sense, is related to one known sequence of inputs and an expected outcome in form of actions or changes in the environment. Since there is integration with a discrete event simulation engine, a test will typically imply: (1) Prepare the AAL service. Initialise simulated devices as expected by the testing sequence (2) Run the simulation. The simulation will start a series of MASON agents which will start controlling their jME avatars and performing actions. During the simulation, MASON agents will interact with the jME environment and, consequently, will affect the network of sensors which is part of the AAL service. Some interactions, like the act of touching a display will be simulated through pre-arranged sequences of events. The AAL service will run in a separated virtual machine, like QEMU in the case of Android smart phones, and will not be controlled by the simulation cycle. (3) Observe simulation variables. The MASON agents do have access to the jME environment of their avatars, so, if they were designed with this intention, these agents ought to have triggers or flags that indicate when their behaviour was successfully performed.

4 Case Study

The case study we focus on for the initial development of PHAT is a classical falling situation. The caregiver is preparing dinner in the kitchen while the patient goes to wash hands in the toilet. At some point of his way, the patient cannot move his feet. The smart phone detects this situation and plays a music which helps the patient to walk again. Once in the toilet, the patient perhaps slips on the water in the ground and falls. The caregiver is in another room and has to, first, listen to the patient fall or help cries, and then run to attend the patient. Figure 2 reproduces this situation. The left window displays the 3D environment which is generated by PHAT where the relative is walking towards the patient. Note that the path that the relative is following is displayed and an obstacle (a water bottle) is in his way. Both windows at right belong to MASON. The upper one is the console where the tab named “Inspectors” shows the properties of the relative. The entities are inspected by double-click on them in

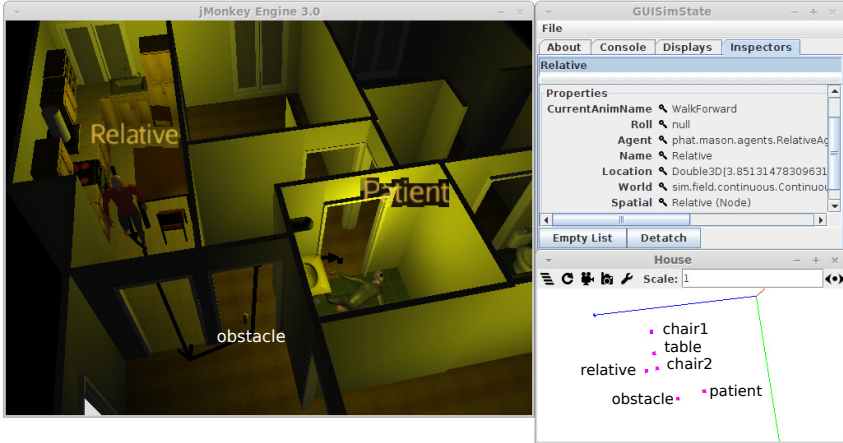


Fig. 2. A capture of PHAT where a patient is falling over

the bottom window named “House”. This window is the MASON world where each physics entity is updated by PHAT using its physics engine.

We make the scenario more complex by making use of JME3 features: (1) Lights. The scenario reproduce a dining situation where the house has its lights off, except the rooms the actors occupy: the kitchen and the toilet. (2) Physics. They are applied to implement the accelerometer sensor of the smart phone and to show a realistic fall of the patient and register with what objects in the house the patient may crash against. In our test, the caregiver stumble upon a water bottle in the ground which is barely visible due to the low light and fall in the corridor. Also, it may happen the caregiver hits the door frame and gets some injure as a result of the collision. (3) Sound. The music from smart phone will helps the patient if she didn’t forgot his device. Also, there is sound coming from the cooking, so it has to be decided if the actor really can listen to the patient.

Actors in the scenario use a combination of physics and animations. When running, the actors use animations since using a real walking algorithm would imply embedded equilibrium mechanisms, limb coordination, and other elements that will be addressed in future releases. In this scenario, the control of actors is defined through automatons running on MASON as follows. (1) Patient. Go to the toilet. Suddenly, a lock event sets that cannot move his legs. After a few seconds, the patient goes on when she hears the music. Once in the toilet, look to another direction, perhaps looking to something else. Then trip over and fall. While in the ground, try to stand up and ask for help. (2) Relative. Do nothing at the beginning. If a noise is heard, then run to the noise source. When reaching the destination, help the caretaker.

For an AAL application, the interest comes from situations where the relative is not aware of the situation (the relative does not hear the fall noise or the cries for help) of the caretaker and/or is unable to provide assistance. By working with the lights, physics, and sound, we can provide variations over the scenario.

5 Conclusions and Future Works

This paper has introduced PHAT, a new framework for simulating living labs for the development of AAL applications. PHAT combines a game engine, jME, with a Multi-Agent Based Simulator, MASON. The result is a simulation that allows to include a great doses of realism. The current test case includes a basic proof of concept of the integration Mason-jME, leaving for jME the environment realisation and the control for MASON. Hence, each MASON agent represents an actor, but it could represent other elements in the scenario as well if a more complex behaviour is needed.

Acknowledgments. This research work has been funded by the Spanish Ministry for Economy and Competitiveness through the project “SOCIAL AMBI-ENT ASSISTING LIVING - METHODS (SociAAL)” (TIN2011-28335-C02-01 and TIN2011-28335-C02-02).

References

1. Garcia-Valverde, T., Campuzano, F., Serrano, E., Villa, A., Botia, J.A.: Simulation of human behaviours for the validation of ambient intelligence services: A methodological approach. *J. Ambient Intell. Smart Environ.* 4(3), 163–181 (2012)
2. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: Mason: A multiagent simulation environment. *Simulation* 81(7), 517–527 (2005)
3. Naranjo, J.-C., Fernandez, C., Sala, P., Hellenschmidt, M., Mercalli, F.: A modelling framework for ambient assisted living validation. In: Stephanidis, C. (ed.) *UAHCI 2009, Part II. LNCS*, vol. 5615, pp. 228–237. Springer, Heidelberg (2009)
4. Sala, P., Kamieth, F., Mocholí, J.B., Naranjo, J.C.: Virtual reality for AAL services interaction design and evaluation. In: Stephanidis, C. (ed.) *Universal Access in HCI, Part III, HCII 2011. LNCS*, vol. 6767, pp. 220–229. Springer, Heidelberg (2011)
5. Campillo-Sanchez, P., Serrano, E., Botía, J.A.: Testing context-aware services based on smartphones by agent based social simulation. *Journal of Ambient Intelligence and Smart Environments* 5, 311–330 (2013)
6. Shoulson, A., Marshak, N., Kapadia, M., Badler, N.I.: Adapt: the agent development and prototyping testbed. In: *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 9–18. ACM, New York (2013)
7. Lewis, M., Jacobson, J.: Game engines in scientific research - introduction. *Commun. ACM* 45(1), 27–31 (2002)
8. Gregory, J.: *Game engine architecture*. Ak Peters Series. A K Peters, Limited (2009)
9. Banks, J., Carson, J.S., Nelson, B.L., Nicol, D.M.: *Discrete-Event System Simulation*, 3rd edn. Prentice Hall (2000)

Support Vector Forecasting of Solar Radiation Values

Yvonne Gala, Ángela Fernández, Julia Díaz, and José R. Dorronsoro

Departamento de Ingeniería Informática and Instituto de Ingeniería del Conocimiento
Universidad Autónoma de Madrid, 28049, Madrid, Spain
yvonne.gala@estudiante.uam.es, {a.fernandez,jose.dorronsoro}@uam.es,
julia.diaz@iic.uam.es

Abstract. The increasing importance of solar energy has made the accurate forecasting of radiation an important issue. In this work we apply Support Vector Regression to downscale and improve 3-hour accumulated radiation forecasts for two locations in Spain. We use either direct 3-hour SVR-refined forecasts or we build first global accumulated daily predictions and disaggregate them into 3-hour values, with both approaches outperforming the base forecasts. We also interpolate the 3-hour forecasts into hourly values using a clear sky radiation model. Here again the disaggregated SVR forecast perform better than the base ones, but the SVR advantage is now less marked. This may be because of the clear sky assumption made for interpolation not being adequate for cloudy days or because of the underlying clear sky model not being adequate enough. In any case, our study shows that machine learning methods or, more generally, hybrid artificial intelligence systems are quite relevant for solar energy prediction.

Keywords: Solar energy, radiation, support vector regression.

1 Introduction

Weather forecasting systems are constantly improving and nowadays provide fairly accurate predictions of the most relevant variables for everyday use. But the high variability of atmospheric phenomena, the very complex nature of Numerical Weather Prediction (NWP) systems and the extremely large underlying systems for the data capture needed to initialize these models make accurate forecasts a very difficult and demanding task, particularly for newer application areas and for the variables relevant for them. A very recent example is solar energy, for which solar radiation is obviously the most relevant variable. However it is affected by phenomena such as cloud behaviour or aerosols, that may not have been too well modelled in current NWP systems or simply are not considered in some such models. This results in NWP radiation forecast accuracies that have to be improved to be successfully applied to effective solar energy predictions, something where Artificial Intelligence techniques can help, either by themselves or as combination of layers of different systems under a hybrid artificial intelligence approach. Of course, the use of statistical or Machine Learning (ML)

methods to post process NWP forecasts is not new. A first example is the regression models proposed in [5] to match solar radiation measures with available forecasts of global horizontal solar radiation (GHR) over horizons from 6 to 30 hours. Similarly, artificial neural networks (ANNs) have been applied in [4] to reduce the relative RMSE of daily average GHR forecasts. See [11] for a review of artificial intelligence techniques in photovoltaic applications.

In this paper we will also follow a ML approach to analyse for two geographic locations in Spain the radiation forecasts of direct horizontal radiation (DHR) given by the European Center for Medium Weather Forecast (ECMWF)¹ as three-hour accumulated radiation values and show how they can be improved after Support Vector Regression (SVR) modelling. We shall consider two different approaches. In the first one we will directly try to derive three-hour radiation predictions applying SVR models to the corresponding three-hour ECMWF forecasts of global horizontal radiation and cloud cover; we will call this approach 3H-SVR models. In the second one we will try to predict total daily radiation for a given day from an input vector given by the three-hour NWP forecasts for that day, the rationale for this is the expectation of the daily radiation problem to be easier, and that better overall results can be disaggregated into better 3-hour accumulated radiation forecasts; we will call this daily approach D-SVR models. Of course, the final goal are hourly radiation forecasts and both 3H-SVR and D-SVR forecasts have also to be disaggregated into hourly values, and we will compare the performance of both approaches and that of ECMWF forecasts. We point out that disaggregating accumulated radiation into hourly values is a difficult problem for which there is not yet an accepted solution. Here we will simply convert three-hour values into hourly forecasts by interpolation using a simple clear sky radiation model. While this may be adequate for cloudless days, it may not be so for cloudy days. In any case, this is an important issue in practical applications that has to be further studied.

The paper is organized as follows. In Sect. 2 we will discuss global horizontal solar radiation, introduce a simple clear sky radiation model which will illustrate the hourly disaggregation procedure that we will follow, and briefly describe Support Vector Regression. In Sect. 3 we will analyse the ECMWF forecasts as well as the daily and 3-hour SVR models we will build to try to improve the base ECMWF forecasts. Moreover, we will describe our disaggregation procedure and compare the hourly performance of the ECMWF and the 3H-SVR and D-SVR models. Finally, Sect. 4 ends the paper with a discussion and pointers to further work.

2 SVR Prediction of Solar Radiation

2.1 Solar Radiation Modelling

As mentioned in the introduction, modelling of solar radiation is a very active research area, where many concrete models have been proposed that often require

¹ European Center for Medium-Range Weather Forecasts. <http://www.ecmwf.int/>

the adjustment of several local parameter values; a review of some of these models from the point of view of renewable energy is in [12]. Since later on the paper we will follow a radiation model approach to disaggregate accumulated radiation values into hourly values, we briefly discuss now a basic, much simplified model of clear sky radiation. Let us denote by \mathcal{I}^d the direct (beam) radiation at a given point, I the horizontal radiation and Θ the solar angle of incidence or zenith angle, that is, the angle between the incident solar rays and the vertical at a point. We then have $I = \mathcal{I}^d \cos \Theta$, where Θ is a trigonometric function $\Theta = \Theta(L, D, H)$ of the latitude L of the point and the day D and hour H .

A first approximation to \mathcal{I}^d could be the solar constant, i.e., the radiation at a given point outside the atmosphere. The actual direct solar radiation at the top of the atmosphere varies with the distance of the Sun to the Earth, with a maximum of about $1,417W/m^2$ at the perihelion, a minimum of about $1,325W/m^2$ at the aphelion and an average value I_A of $1,370W/m^2$. If N denotes the day number, a good approximation for the direct solar radiation I_S is [6]

$$I_S = 1,370 \left(1 + 0.034 \cos \left(2\pi \frac{N - 3}{365} \right) \right);$$

recall that the aphelion falls approximately on January 3. However, to derive \mathcal{I}^d we have to take into account the length of the path the Sun rays follow in the atmosphere. This relative length is described in terms of the air mass A . When the zenith angle Θ is 0 and the Sun is directly overhead, air mass is taken as 1. When Θ increases so does path length and, hence, air mass. The empirical Kasten–Young formula [7] gives an approximation to A

$$A = \frac{1}{\cos \Theta + 0.50572(96.07995 - \Theta)^{-1.6364}}$$

with Θ given here in degrees. A simple formula to combine air mass and the solar constant to derive the direct radiation \mathcal{I}^d is the one proposed by Meinel [10]

$$\mathcal{I}^d = 1.353 \times 0.7^{A^{0.678}}.$$

Putting together these formulæ one can get an approximation to clear sky direct horizontal radiation I that ultimately depends only on Θ , that is, latitude, day and hour (although many other parameters have to be set that are heavily dependent on local conditions). The model described is just a simple clear sky model; in Sect. 3 we will use the more sophisticated Bird model [1].

In any case, these models are a first step towards the highly sophisticated methods used by state of the art numerical weather prediction (NWP) systems. Among them the already mentioned global ECMWF model is very widely used and considered among the best globally. Its forecasts are available twice a day on a $0.25^\circ \times 0.25^\circ$ grid and have a temporal resolution of three hours for the first forecast days. In this study we will work with two variables, three-hour aggregated downward surface solar radiation (DSSR) and average total cloud cover (TCC). However, factors such as the complexity of cloud physics or even the

wide spatial resolution make radiation rather difficult to predict by the ECMWF or other NWP models. As a consequence, NWP models have been shown to have prediction biases and, therefore, limited forecast accuracy. This has been reported in many studies, such as those in [14,3], not only for ECMWF but also for several other NWP systems [8]. Model Output Statistics (MOS) is a technique widely used in meteorology that applies statistical analysis to correct NWP bias and yield refined forecasts. MOS has been applied to radiation estimates under different approaches [8,2,13,9]. In some sense the application of SVR that we will present in Sect. 3 can also be seen as an alternative approach to MOS. Before that, we give next a quick overview of SVR.

2.2 Support Vector Regression

Linear Support Vector Regression (SVR) [15] tries to fit a linear model $W \cdot X + b$ to a sample $\mathcal{S} = \{(X_p, y_p) : p = 1, \dots, N\}$ so that the following criterion function is minimized

$$\min_{W, b, \xi} \frac{1}{2} \|W\|^2 + C \sum_i (\xi_i + \xi_i^*), \quad (1)$$

subject to the restrictions $W \cdot X_i + b - y_i \geq -\xi_i - \epsilon$, $W \cdot X_i + b - y_i \leq \xi_i^* + \epsilon$, $\xi_i, \xi_i^* \geq 0$. The problem (1) represents an extension of standard SVM classification but, in fact, it can be rewritten as a more familiar modelling problem. First, notice that we only penalize strictly positive ξ_i and ξ_i^* slacks. Moreover, observe that if $W \cdot X_i + b - y_i > 0$, we would have $|W \cdot X_i + b - y_i| \leq \xi_i^* + \epsilon$, while when $W \cdot X_i + b - y_i < 0$ we would have $|W \cdot X_i + b - y_i| \leq \xi_i + \epsilon$. Thus, minimizing (1) is equivalent to minimize

$$\sum_i [y_i - W \cdot X_i - b]_\epsilon + \frac{1}{C} \|W\|^2,$$

where $[z]_\epsilon$ is the ϵ -insensitivity cost function $[z]_\epsilon = \max(0, |z| - \epsilon)$. In other words, SVR can be seen as a variant of standard L_2 regularized regression where instead of the familiar $z_i^2 = (y_i - W \cdot X_i - b)^2$ square error, we use the $[z_i]_\epsilon$ errors that allow an ϵ -wide, penalty-free “error tube” around the model function $W \cdot X + b$. In any case, what one actually solves is the dual of (1), that is, minimizing

$$J(\alpha, \beta) = \frac{1}{2} \sum_{i,j} (\alpha_i - \beta_i)(\alpha_j - \beta_j) X_i \cdot X_j + \epsilon \sum_i (\alpha_i + \beta_i) - \sum_i y_i (\alpha_i - \beta_i) \quad (2)$$

where now $0 \leq \alpha_i, \beta_i \leq C$. In our experiments we will solve (2) using the LIBSVM implementation ² which can be considered the state of the art in

² LIBSVM. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

SVM software. Moreover, notice that (2) only involves dot products $X_i \cdot X_j$ and the same is true for the SMO algorithm used in LIBSVM implementation; in particular, we can replace these dot products using a kernel k so that, instead of the X_i , we work with a high (possibly infinite) dimensional projection $\Phi(X_i)$ so that $\Phi(X_i) \cdot \Phi(X_j) = k(X_i, X_j)$. Here we will use a Gaussian kernel $k(X, X') = \exp\left(-\frac{\|X-X'\|^2}{2\sigma^2}\right)$.

Once we solve (2) and obtain the optimal α_i^*, β_i^* , the optimal weight is now $W^* = \sum (\alpha_i^* - \beta_i^*) \Phi(X_i)$ and the final model is given by

$$\begin{aligned} f(X) &= b^* + W^* \cdot \Phi(X) = b^* + \sum \gamma_i^* \Phi(X_i) \cdot \Phi(X) = b^* + \sum_i \gamma_i^* K(X_i, X) \\ &= b^* + \sum_i \gamma_i^* e^{-\frac{\|X-X_i\|^2}{2\sigma^2}}. \end{aligned}$$

where we write $\gamma_i^* = \alpha_i^* - \beta_i^*$. We describe next how to build and apply SVR models to forecast radiation values.

3 Experiments

3.1 ECMWF Radiation Forecasts

We shall compare first here ECMWF DHR forecasts and actual measured values at two concrete points in Spain, A Coruña and Alicante, for which there are hourly DHR measures. The data for the study go from December 2009 to June 2011, that is, 19 months. Since in the next section we will work with ML models that require a certain time period to be used as the training data to build models, we will consider a subset of the above months as a test period, namely, from January in 2011 to June in 2011. Actual radiation values for the two locations are given for solar hours 5 to 21. On the other hand, ECMWF forecast of DHR are given as 3-hour accumulated values for hours 6, 9, 12, 15, 18 and 21. Summer accumulated radiation values at solar hour 6 certainly include radiation at hour 5 and even a slight amount at hour 4. We do not have actual radiation values at hour 4, but in what follows we will ignore this and consider as 3-hour accumulated radiation at hour 6 the sum of radiations at hours 5 and 6. Therefore, we will first compare 3-hour actual accumulated radiation for solar hours 6, 9, 12, 15, 18 and 21 with the corresponding ECMWF forecasts, and measure the monthly mean absolute error, MAE, for both the individual 3-hour forecasts and the total daily radiation. More precisely, denoting by $I_{d,h}^3$ the accumulated 3-hour radiation up to hour h of day d and by $\widehat{I}_{d,h}^{3,E}$ its ECMWF prediction, the 3-hour ECMWF mean absolute error (MAE) $e_{3H}^E(m)$ at month m is given by

$$e_{3H}^E(m) = \frac{1}{ND_m} \sum_{d=1}^{ND_m} \frac{1}{6} \sum_{k=2}^7 |I_{d,3k}^3 - \widehat{I}_{d,3k}^{3,E}|,$$

with ND_m the number of days in month m .

For total daily radiation forecasts, we transform the accumulated 3-hour radiation ECMWF forecasts $\widehat{I}_{d,h}^{3;E}$ into daily forecasts \widehat{I}_d^E as $\widehat{I}_d^E = \sum_{k=2}^7 \widehat{I}_{d,3k}^{3;E}$. Similarly as just done, denoting by I_d the total radiation for day d , i.e. $I_d = \sum_{h=5}^{21} I_{d,h}^3$, the daily ECMWF forecasts error $e_D^E(m)$ at month m is given by

$$e_D^E(m) = \frac{1}{ND_m} \sum_{d=1}^{ND_m} |I_d - \widehat{I}_d^E|.$$

Finally, considering hourly radiation values $I_{d,h}$ for hour h of day d and the corresponding ECMWF forecasts $\widehat{I}_{d,h}^E$ (we discuss below how we will derive them), the hourly ECMWF MAE error $e_H^E(m)$ for a month m is given by

$$e_H^E(m) = \frac{1}{ND_m} \sum_{d=1}^{ND_m} \frac{1}{17} \sum_{h=5}^{21} |I_{d,h} - \widehat{I}_{d,h}^E|.$$

Tables 1, 2 and 4 give the errors of the 3-hour, daily and hourly ECMWF forecast for the test months.

Table 1. 3-hour MAE errors of ECMWF and SVR forecasts

	A Coruña						
	Jan	Feb	Mar	Apr	May	Jun	Ave
ECMWF	39.06	50.57	66.58	71.19	79.47	88.64	65.92
D-SVR	34.29	39.68	51.45	66.51	79.30	84.03	59.21
3H-SVR	34.22	42.01	52.37	64.03	72.92	87.00	58.76
	Alicante						
	Jan	Feb	Mar	Apr	May	Jun	Ave
ECMWF	29.21	33.81	57.73	47.71	53.09	50.28	45.30
D-SVR	28.67	32.47	54.60	44.04	53.30	51.79	44.15
3H-SVR	27.42	30.51	58.12	45.09	52.03	50.16	43.89

3.2 SVR Radiation Forecasts

We will consider two different SVR approaches to forecast DHR values. On the one hand, we will first directly model 3-hourly accumulated DHR values (3H-SVR), using as targets the accumulated values of the measured DHR, i.e., $I_{d,3k}^3 = I_{d,3k} + I_{d,3k-1} + I_{d,3k-2}$, $k = 2, \dots, 7$.

As input patterns $X_{d,h}$ we will take the DSSR and TCC forecasts at each one of the four grid corners that surround the measurement point; pattern dimension is thus 8. These 3-hour forecasts $\widehat{I}_{d,h}^{3;3S}$ can be added to derive a daily radiation forecast $\widehat{I}_d^{3S} = \sum_2^7 \widehat{I}_{d,3k}^{3;3S}$.

On the other hand, we can build D-SVR models that aim to forecast the total daily DHR I_d from patterns made up of the six 4-point DSSR and TCC forecasts

Table 2. Daily total MAE errors of ECMWF and SVR forecasts

A Coruña							
	Jan	Feb	Mar	Apr	May	Jun	Ave
ECMWF	151.90	198.99	250.70	227.03	272.79	357.13	243.09
D-SVR	118.41	126.67	146.78	237.07	251.47	342.93	203.89
3H-SVR	126.60	132.80	169.79	246.55	262.27	370.27	218.05
Alicante							
	Jan	Feb	Mar	Apr	May	Jun	Ave
ECMWF	127.96	103.88	284.56	233.82	246.72	233.26	205.03
D-SVR	124.30	102.43	238.78	182.10	205.16	226.00	179.79
3H-SVR	107.58	93.99	256.94	181.02	195.41	221.17	176.02

given by the ECMWF; pattern dimension is now $6 \times 8 = 48$. To disaggregate the daily D-SVR forecasts \hat{I}_d^{DS} into 3-hour accumulated DHR forecasts $\hat{I}_{d,h}^{3;DS}$ we will use the proportions determined by the ECMWF radiation forecasts, i.e., we have

$$\hat{I}_{d,3k}^{3;DS} = \frac{\hat{I}_{d,3k}^{3;E}}{\hat{I}_d^E} \hat{I}_d^{D,DS},$$

where we recall that $\hat{I}_d^E = \sum_{k=2}^7 \hat{I}_{d,3k}^{3;E}$.

In order to build a 3H-SVR or D-SVR model for each one of the testing months, observe that for each month m we have to select the optimal penalty factor C , tolerance ϵ and Gaussian kernel width σ of the corresponding SVR models. There are several options that can be considered and, in fact, given the clear seasonal behaviour of radiation, the adequate selection of the validation and training subsets may have a great influence on a model's performance, as a good model for, say, August, will not give good results when tested in, say, January. We will work with a sliding 12-month training set and use for validation the month just prior to the one to be used for testing. In other words, to build a model for month m , we use as validation subset month $m - 1$ and months $m - 13$ to $m - 2$ as the training set.

We will use the LIBSVM code to construct the SVR models that also implements exhaustive grid search to choose the best parameters. For the penalty terms C_D and C_{3H} we consider values in the range $[2^7, 2^{15}]$ with a step of 2^1 ; the range for σ_D and σ_{3H} is $[2^{-18}, 2^3]$ with a step of 2^2 and the range for ϵ_D and ϵ_{3H} is $[2^{-1}, 2^8]$ with a step of 2^1 .

Table 3 shows the best parameters for each monthly model for Alicante and A Coruña.

The errors of the accumulated 3-hour and daily SVR forecasts are also given in Tables 1 and 2, respectively. At first sight, we could expect the 3H-SVR forecasts to be better than the ECMWF and D-SVR ones for 3-hour forecasts, while D-SVR should be better for daily forecasts. Looking at the average MAE errors, both 3H-SVR and D-SVR improve on the ECMWF forecasts. 3H-SVR

Table 3. Best parameters for each monthly model for Alicante and A Coruña

	A Coruña						Alicante					
	C_D	C_{3H}	σ_D	σ_{3H}	ϵ_D	ϵ_{3H}	C_D	C_{3H}	σ_D	σ_{3H}	ϵ_D	ϵ_{3H}
Jan	2^{12}	2^{15}	2^{-9}	2^{-13}	2^8	2^{-1}	2^{14}	2^7	2^{-13}	2^{-7}	2^1	2^2
Feb	2^{12}	2^9	2^{-9}	2^{-7}	2^7	2^4	2^9	2^9	2^{-7}	2^{-7}	2^{-1}	2^{-1}
Mar	2^8	2^{11}	2^{-7}	2^{-5}	2^5	2^2	2^{14}	2^8	2^{-13}	2^{-11}	2^7	2^3
Apr	2^7	2^{15}	2^{-5}	2^{-5}	2^6	2^5	2^{14}	2^{15}	2^{-11}	2^{-7}	2^7	2^5
May	2^9	2^{12}	2^{-7}	2^{-3}	2^7	2^5	2^{11}	2^{14}	2^{-5}	2^{-3}	2^4	2^5
Jun	2^{15}	2^9	2^{-13}	2^{-1}	2^8	2^5	2^{14}	2^{15}	2^{-13}	2^{-7}	2^7	2^{-1}

is better for the 3-hour forecasts, and D-SVR is better for daily forecasts in A Coruña, although 3H-SVR slightly beats it for Alicante. As it can be seen, both the daily and 3H monthly predictions improve on the ECMWF forecasts except for April in A Coruña.

3.3 Hourly Disaggregation of Radiation Forecasts

A clear sky model gives the DHR value at time t of day d as a function $c(t; d)$. Thus, the accumulated radiation between hours $h - 1$ and h of day d is

$$a(h; d) = \int_{h-1}^h c(t; d) dt \simeq \frac{1}{M} \sum_{i=1}^M c\left(h - 1 + \frac{i}{M}; d\right),$$

where M determines the approximation time step. To disaggregate the 3-hour accumulated DHR $\widehat{I}_{d,h}^3$ as the sum $\widehat{I}_{d,h}^3 = \widehat{I}_{d,h} + \widehat{I}_{d,h-1} + \widehat{I}_{d,h-2}$ of the hourly DHR values, a simple approach to derive the hourly forecasts $\widehat{I}_{d,3k-2}, \widehat{I}_{d,3k-1}, \widehat{I}_{d,h}$ from $\widehat{I}_{d,3k}^3$ is

$$\widehat{I}_{d,3k-j} = \frac{a(3k - j; d)}{A(3k; d)} I_{d,3k}^3,$$

where $k = 2, \dots, 7$ and we take $A(h; d) = a(h - 2; d) + a(h - 1; d) + a(h; d) = \int_{h-3}^h c(t; d) dt$. Notice that this ensures that $\widehat{I}_{d,3k}^3 = \widehat{I}_{d,3k} + \widehat{I}_{d,3k-1} + \widehat{I}_{d,3k-2}$.

We will apply this approach to derive hourly radiation forecasts from the ECMWF, 3H-SVR and D-SVR 3-hour forecasts $\widehat{I}_{d,h}^{3,E}, \widehat{I}_{d,h}^{3,3S}$ and $\widehat{I}_{d,h}^{3,DS}$ respectively using the previously mentioned Bird model [1] for the function $c(t; d)$. The errors of the disaggregated hourly radiation forecasts of the ECMWF, 3H-SVR and D-SVR models are given in Table 4. Here we could expect that 3H-SVR, the best 3-hour forecasts, would also lead to the best hourly ones and this is so for the average and monthly MAEs. However, errors are now tighter than for the 3H and D cases, suggesting that the hourly interpolation should be improved. One source of this tightening is that the assumption of 3-hour values decomposing according to clear sky interpolation may not be correct for some days (such as, for instance, cloudy ones). Moreover, we recall that clear sky models require

Table 4. Hourly MAE errors of ECMWF and SVR forecasts

	A Coruña						
	Jan	Feb	Mar	Apr	May	Jun	Ave
ECMWF	19.55	22.92	29.28	32.64	37.74	39.17	30.22
D-SVR	17.88	19.11	24.54	31.39	37.78	37.29	28.00
3H-SVR	17.71	19.85	24.56	30.72	35.11	38.31	27.71
	Alicante						
	Jan	Feb	Mar	Apr	May	Jun	Ave
ECMWF	17.80	17.66	27.44	24.36	27.33	24.68	23.21
D-SVR	16.84	17.53	26.03	22.90	27.46	25.87	22.77
3H-SVR	16.57	16.80	27.43	23.14	26.66	24.85	22.57

several parameters to be locally adjusted and it may be very well the case that the parametrization of the Bird model used is not optimal for the A Coruña and Alicante locations.

4 Discussion and Conclusions

The increasing integration of renewable energies into the electrical system is demanding better forecasts of the meteorological variables involved. Of course this requires advances in the NWP systems now in exploitation but their complexity makes these advances relatively slow. In the mean time, hybrid artificial intelligence systems can be of great interest and in this work we have shown how ML techniques, SV regression here, can improve the initial radiation forecasts provided for two locations in Spain by the state-of-the-art ECMWF model. More precisely, our results show that clearly more accurate local forecasts can be derived applying SVR models that have as inputs the ECMWF predictions.

An obvious conclusion is thus that SVR or other ML methods may be exploited with profit to yield better forecasts in the field of renewable energies. While we have not followed this approach here, our results suggest to apply ML models to obtain actual energy production forecasts that may result in improved economic yields to farm operators (as they will have to bear smaller deviation penalties) and an easier operation and integration of these energies in the electrical system. In any case, there is clearly room for further improvements, particularly when the SVR-refined 3-hour accumulated radiation forecasts have to be disaggregated into hourly values, as the clear advantages of SVR forecasts over the ECMWF ones for 3-hour or daily accumulated radiation become less strong. Reasons for this may be the clear sky interpolation approach followed here or, even, the underlying clear sky model used. Although there seems to be no universally accepted best clear sky model (and they also need careful local calibration of model parameters), using more modern models (see [9]) would certainly help. Another possibility is to, first, disaggregate the 3-hour NWP forecasts into hourly values and then use them to model directly hourly radiation values. We are presently looking into these and other related issues.

Acknowledgement. With partial support from Spain’s grant TIN2010-21575-C02-01 and the UAM–ADIC Chair for Machine Learning. The first author is also supported by an UAM–ADIC Chair grant and the second author by an FPI-UAM grant. We thank the Departamento de Aplicaciones para la Operación of Red Eléctrica de España for providing the radiation data and many helpful discussions.

References

1. Bird, R.E., Hulstrom, R.: Simplified clear sky model for direct and diffuse insolation on horizontal surfaces. Tech. Rep. No. SERI/TR-642-761, Solar Energy Research Institute, Golden, CO (1981)
2. Bofinger, S., Heilscher, G.: Solar electricity forecast—approaches and first results. In: 21st European Photovoltaic Solar Energy Conference (2006)
3. Espinar, B., Ramires, L., Drews, A., Beyer, H.G., Zarzalejo, L.F., Polo, J., Martin, L.: Analysis of different comparison parameters applied to solar radiation data from satellite and german radiometric stations. *Solar Energy* 83(1), 118–125 (2009)
4. Guarneri, R., Martins, F., Pereira, E., Chuo, S.: Solar radiation forecasting using artificial neural networks. *National Institute for Space Research* 1, 1–34 (2008)
5. Jensenius, J., Cotton, G.: The development and testing of automated solar energy forecasts based on the model output statistics (MOS) technique. In: 1st Workshop on Terrestrial Solar Resource Forecasting and on Use of Satellites for Terrestrial Solar Resource Assessment, pp. 22–29 (1981)
6. Kandilli, C., Ulgen, K.: Solar illumination and estimating daylight availability of global solar irradiance. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 30, 1127–1140 (2008)
7. Kasten, F., Young, A.: Revised optical air mass tables and approximation formula. *Applied Optics* 38, 4735–4738 (1989)
8. Lorenz, E., Remund, J., Müller, S., Traunmüller, W., Steinmaurer, G., Ruiz-Arias, J., Fanego, V., Ramirez, L., Romeo, M., Kurz, C., Pomares, L., Guerrero, C.: Benchmarking of different approaches to forecast solar irradiance. In: 24th European Photovoltaic Solar Energy Conference, pp. 4199–4208 (2009)
9. Mathiesen, P., Kleissl, J.: Evaluation of numerical weather prediction for intra-day solar forecasting in the continental united states. *Solar Energy* 85, 967–977 (2011)
10. Meinel, A.B., Meinel, M.P.: *Applied Solar Energy*. Addison Wesley Publishing Co. (1976)
11. Mellit, A., Kalogirou, S.A.: Artificial intelligence techniques for photovoltaic applications: A review. *Progress in Energy and Combustion Science* 34(5), 574–632 (2008)
12. Myers, D.R.: Solar radiation modeling and measurements for renewable energy applications: data and model quality. *Energy* 30(9), 1517–1531 (2005)
13. Perez, R., Kivalov, S., Schlemmer, J., Hemker Jr., K., Renné, D., Hoff, T.E.: Validation of short and medium term operational solar radiation forecasts in the US. *Solar Energy* 84(12), 2161–2172 (2010)
14. Remund, J., Perez, R., Lorenz, E.: Comparison of solar radiation forecasts for the USA. In: *European PV Conference, Valencia, Spain* (2008)
15. Schölkopf, B., Smola, A.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press (2001)

A Hybrid Fuzzy Approach to Facility Location Decision-Making

Dragan Simić^{1,*}, Vasa Svirčević², and Svetlana Simić³

¹University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6,
21000 Novi Sad, Serbia
dsimic@eunet.rs

²Lames Ltd., Jarački put bb., 22000 Sremska Mitrovica, Serbia
vasasv@hotmail.com

³University of Novi Sad, Faculty of Medicine, Hajduk Veljkova 1–9, 21000 Novi Sad, Serbia
drdragansimic@gmail.com

Abstract. Facility location decisions play a critical role in the strategic design of supply chain networks. Selection of facility locations among alternative locations is a decision problem which includes quantitative and qualitative criteria simultaneously. This paper discusses facility location problem with focus on logistics distribution center in Novi Sad area, Serbia, micro-location selection. Methodological fuzzy TOPSIS ranking method is proposed and examined and it is shown how such a model can be of assistance in analyzing a multi criteria decision-making problem when the information available is vague and subjective. The experimental results could be compared with other official results of the feasibility study of the distribution center (DC) located in Novi Sad area. Compared to the official study, which does not show a methodological basis, this research gets the results similar to the empirical results in an entirely exact way.

Keywords: Facility location, logistics distribution center, fuzzy logic, TOPSIS.

1 Introduction

Facility location decisions play a critical role in the strategic design of supply chain networks. Facility location problem, whose optimization is a central area in operations research, is determining the best region for facility. Typical application of facility location includes placement of factories, warehouses, storage facilities, depots, schools, libraries, fire stations, hospitals, ATM machines, base stations for wireless services. Selection of facility locations among alternative locations is a decision problem which includes quantitative and qualitative criteria simultaneously.

It is possible to consider the facility location problem with focus on logistics distribution center (LDC) in Novi Sad area, Serbia, micro-location selection, in different ways. During the decision phase, it is possible to, in a first place, define two major category groups: The first major category group is based on belonging to a certain area and is divided into six categories which are: technology, economics, organizational, technical,

ecological, legal and regulatory. Then the second major category group represents business interest groups such as: user system, system of terminals, social system. There are, all in all, 69 criteria that should be taken into account when choosing optimal location for LDC in Novi Sad area. This paper discusses only first major category group and its six categories with appropriate 38 criteria.

The rest of the paper is organized in the following way: Section 2 provides some approaches about facility location selection methods and related work. In Section 3 Official Study selecting LDC in Novi Sad area is presented. Section 4 proposes methodological fuzzy TOPSIS ranking method, which is examined and it is shown how such a model can assist in analyzing a multi criteria decision-making problem when the information available is vague and subjective. Selection of a facility for LDC, categories, criteria and facility rating by decision makers with respect to established categories are discussed in Section 5. An application of this hybrid model's experimental result, the Euclidean distance measure and rank, the order of the facility location and selection of LDC in Novi Sad area, are presented in Section 6. Finally, Section seven gives concluding remarks.

2 Facility Location Problem and Related Work

One of the more important discussions in production planning is facility planning, in which facility location and layout are considered. Since the decision makers have different opinions about importance and relations between facilities and locations, a multi-criteria analysis helps them to solve problem in group decision-making process.

Different researches have discussed location problems with multiple attributes and various methods are used for solving them. In [1] a fuzzy goal programming for locating a single facility with a given convex region is presented. Solving facility location problems using different solution approaches of fuzzy multi-attribute group decision making is presented in [2]. In [3] fuzzy AHP and fuzzy TOPSIS are used for the selection of facility location. In that particular research they compare results of the proposed methods. Fuzzy TOPSIS is proposed for selecting plant location in [4]. A closeness coefficient is defined to determine the ranking order of alternative locations by calculating the distances to both ideal and negative solutions. A new TOPSIS approach for selecting plant location under linguistic environments, where the ratings are various alternative locations under various criteria, and the weights of various criteria are assessed in linguistic terms represented by fuzzy number presented in [5]. A new fuzzy multiple-attribute decision-making approach, for solving facility location selection problems by using objective-subjective attributes is presented in [6]. The proposed system integrates fuzzy set theory, factor rating system, and simple additive weighting (SAW) and it is applied to deal with both qualitative and quantitative dimensions [6].

A method to determine the optimal location of fire station facilities is discussed in [7]. The model is a combination of a fuzzy multi-objective programming and a genetic algorithm. The objectives are to minimize the total setup and operating costs of fire stations and total loss cost of accidents in a given area and minimize the longest

distance from a fire station to any accident site. The fuzzy multi-objectives are appropriately converted to a single unified min-max goal for being solved by a genetic algorithm [7].

The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) method, which was initially proposed in [8], is a well-known multiple criteria decision making (MCDM) method. The TOPSIS method introduces the shortest distance from positive ideal solution (PIS) and the farthest distance from the negative ideal solution (NIS) to determine the best alternative. While the PIS is maximal benefit criteria and NIS is minimal benefit criteria.

TOPSIS method has become popular multiple criteria decision technique due to (1) its theoretical rigorousness [9], (2) a sound logic that represents the human rationale in selection [10], and (3) the fact that it has been provided in [11] as one of the most appropriate methods in solving traversal rank. Recently, some researchers have focused on developing fuzzy TOPSIS methods to deal with imprecise information. Sun [12] applied fuzzy TOPSIS to evaluate the competitive advantage of shopping websites. Kahraman [13] proposed an interactive group decision making methodology based on fuzzy TOPSIS method to select information system providers under multiple criteria.

3 Choosing a Location for a Distribution Center – Official Study

Based on an existing study: „Studija o uslovima i opravdanosti izgradnje RTC u Novom Sadu (A study on the conditions and justification of construction of the freight transport center in Novi Sad)” [14], done by PU ”Urban Planning”– Institute of urban planning Novi Sad in 2004. Presents certain locations in the town which could be potential locations for future distribution center. Our research relies on certain elements of the above mentioned study, so processing of these potential sites with their advantages and disadvantages confirms or denies the final site selection for the development of distribution center.

The above mentioned study offers an analysis of the current status of all activities involved in the operation of a distribution center, starting with the transportation systems, freight transport, the size of sub-systems, distribution center, the size of the overall space required. The analysis determines the criteria for site selection, as well as decision on the wider range of sites, and finally selection of a site based on the following criteria: (1) The relationship of the site to the General Plan of Novi Sad up to the 2021., (2) relationship to the obligations based on existing plans and zoning documents issued, (3) spatial and physical suitability: characteristics of natural conditions, the condition of the physical structure, land use, infrastructure, transport facilities, environmental protection requirements.

Location 1. located in an area that is, by the General Plan, intended to be a work zone. It's part of the work zone, *North III* that includes complex of Novi Sad port, the Free Zone, companies *Danubius* and *Agrohem*. The total surface area is about 53 ha.

Location 2 is located in the commercial zone, *North IV* on the part of a space used for regional port with a surface area of about 190 ha.

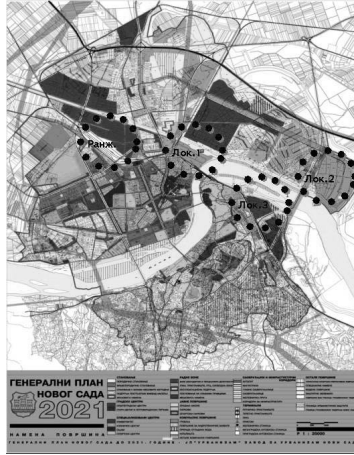


Fig. 1. Potential distribution center locations considered in the Official Study [14]

Location 3 located within the work zone *East*. Of the gross surface area of about 217 hectares, the company *Pobeda-holding* has taken up 41 ha, *Rokov potok* and corridors occupy about 82 ha, and the settlement *Sadovi* take up area of 9.6 ha.

4 A Fuzzy TOPSIS Method for Facility Location Selection

The decision matrix, R_t , given by group of k decision makers (D_1, D_2, \dots, D_k), d_t , $t = 1, 2, \dots, k$, for ranking m alternatives facility location (L_1, L_2, \dots, L_m) with respect to n decision categories (C_1, C_2, \dots, C_n):

$$R_t = \begin{bmatrix} r_{11t} & r_{12t} & \dots & r_{1nt} \\ r_{21t} & r_{22t} & \dots & r_{2nt} \\ \dots & \dots & \dots & \dots \\ r_{m1t} & r_{m2t} & \dots & r_{mnt} \end{bmatrix} \quad (1)$$

The $r_{ijt} = (l_{ijt}, c_{ijt}, d_{ijt})$, $r_{ijt} \in R^+$, $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$; $t = 1, 2, \dots, k$ are used to denote the rating of alternative l_i with respect to decision categories c_j given by the d_t . The procedure of the fuzzy TOPSIS method starts in the following way.

Step 1. Aggregate the importance weights. Let $w_{jt} = (l_{ijt}, c_{ijt}, d_{ijt})$, $j = 1, 2, \dots, n$; $t = 1, 2, \dots, k$; be the importance weight of decision categories C_j given by decision maker d_t . Then it can be calculated the aggregated crisp weight W_j of categories C_j by the following, where w'_{jt} is the weight derived from the graded mean integration representation of fuzzy numbers, as illustrated in Equation (2).

$$W_j = \frac{\sum_{t=1}^k w'_{jt}}{k} \quad (2)$$

Step 2. Aggregate rating of alternatives. The following formula is used to obtain the aggregated crisp rating of alternatives R_{ij} .

$$R_{ij} = \frac{\sum_{t=1}^k r'_{ijt}}{k} \quad (3)$$

where r'_{ijt} is obtained by the graded mean integration representation of fuzzy numbers.

Step 3. Construct normalized and weighted decision matrix. Let $S = [s_{ij}]_{m \times n}$ be the normalized decision matrix. It can be calculated as the normalized value s_{ij} by the following formula.

$$s_{ij} = \frac{R_{ij}}{\sqrt{\sum_{i=1}^m (R_{ij})^2}} \quad (4)$$

Let $V = [v_{ij}]_{m \times n}$ be the weighted decision matrix. The weighted value v_{ij} is derived from the product of elements in the normalized decision matrix and weights.

$$v_{ij} = W_j s_{ij} \quad (5)$$

Step 4. Determine the Positive Ideal Solution (PIS) and the Negative Ideal Solution (NIS). Let I and J be the index sets associated with the alternative set and the criterion set, respectively. There can gain the PIS, A^+ , and NIS, A^- , from the following methods.

$$A^+ = \{v_1^+, v_2^+, \dots, v_n^+\} = \left\{ \max_{i \in I} v_{ij} \mid j \in J \right\} \quad (6)$$

$$A^- = \{v_1^-, v_2^-, \dots, v_n^-\} = \left\{ \max_{i \in I} v_{ij} \mid j \in J \right\} \quad (7)$$

Step 5. Measure the distance of each facility location alternatives from the PIS and NIS respectively. Traditionally, the Euclidean distance is used to measure distance of each alternative from A^+ and A^- as follows.

$$d_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_{ij}^+)^2}, i=1, 2, \dots, m \quad (8)$$

$$d_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_{ij}^-)^2}, i=1, 2, \dots, m \quad (9)$$

However, there may be a problem in the use of the Euclidean distance associated with weight having been calculated twice. This problem can be resolved by introducing Eq. (10) or Eq. (11) as follows.

$$d_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_{ij}^+)^2} = \sqrt{\sum_{j=1}^n (W_j s_{ij} - W_j s_j^+)^2} = \sqrt{\sum_{j=1}^n W_j^2 (s_{ij} - s_j^+)^2} \quad (10)$$

Therefore, this problem can be overcome by means of Minkowski distance, L_p^w , as follows.

$$L_p^w(x, y) = \left[\sum_{j=1}^n w_j |x_j - y_j|^p \right]^{1/p} \quad (11)$$

where w_j is the weight of importance with respect of the j -th criterion and $p \geq 1$.

Note that L_p^w with $p=2$ is known as the weighted Euclidean distance. Based on the weighted Euclidean distance, A^+ and A^- can be redefined in the following way. Recall that $S=[s_{ij}]$ is the normalized decision matrix. Define

$$A^+ = \{s_1^+, s_2^+, \dots, s_n^+\} = \left\{ \left(\max_{i \in I} s_{ij} \mid j \in J \right) \right\} \quad (12)$$

$$A^- = \{s_1^-, s_2^-, \dots, s_n^-\} = \left\{ \left(\min_{i \in I} s_{ij} \mid j \in J \right) \right\} \quad (13)$$

and then the distance of each alternative from A^+ and A^- based on the weighted Euclidean distance is computed as

$$d_i^+ = \sqrt{\sum_{j=1}^n W_j |s_{ij} - s_j^+|^2}, i=1, 2, \dots, m \quad (14)$$

$$d_i^- = \sqrt{\sum_{j=1}^n W_j |s_{ij} - s_j^-|^2}, i=1, 2, \dots, m \quad (15)$$

Step 6. Calculate the relative closeness coefficient and rank of the performance order. The relative closeness coefficient of the i -th facility location alternative, RCC_i , can be computed by

$$RCC_i = \frac{d_i^-}{d_i^+ + d_i^-} \quad (16)$$

Consequently, the alternatives can be ranked according to RCC_i .

5 Selecting Facility in Novi Sad Location

A group of three decision makers, D_t , $t = 1, \dots, 3$, has been formed for evaluation made to conduct the assessment based on six categories which include 38 criteria, denoted by C_l , $l = 1, \dots, 6$. Shows the hierarchical structure for facility location problem.

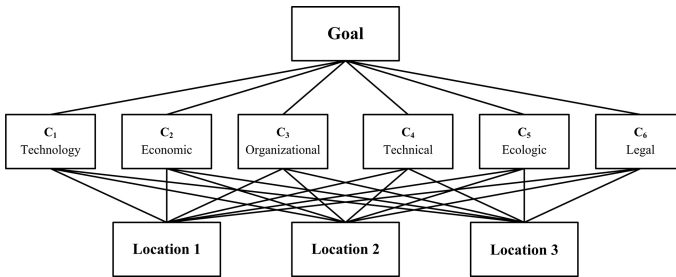


Fig. 2. A hierarchical structure for facility location problem

The details for these group criteria are listed in the following way: (1) Technology category (C_1): the intensity of traffic flows (c_{11}), terminal availability (c_{12}), distance from the user (c_{13}), time of delivery (c_{14}), availability of technology and the type of commodity (c_{15}), connections with other modes of transport (c_{16}), availability of the terminal intermodal transport (c_{17}); (2) Economic category (C_2): logistics costs (c_{21}), site activation costs (c_{22}), investment in the construction of access roads and infrastructure (c_{23}), net present value of the site (c_{24}), intern profitability rate (c_{25}), return on investment period (c_{26}), gravitating towards economically developed economy (c_{27}); Organizational category (C_3): logistics provider presence category (c_{31}), intermodal transport operator presence (c_{32}), the possibility of the line connection with the rail and water transport (c_{33}), offices, associations in the field of transport and logistics (c_{34}); Technical category (C_4): location geological characteristic (c_{41}), infrastructure network (c_{42}), technical possibilities of connection with the traffic infrastructure (c_{43}); Ecologic category (C_5): air pollution (c_{51}), noise and vibration (c_{52}), hazardous materials (c_{53}), dangerous goods (c_{54}), environmental impact of the goods in the terminal (c_{55}), impact of goods in process in the terminal on the environment (c_{56}); Legal and regulatory category (C_6): harmonization with spatial and urban planning (c_{61}) the possibility of regulating the ownership of land and buildings (c_{62}), complying with the law (c_{63}); dangerous goods (c_{64}).

Table 1. Rating by DMs with respect to categories

Cat.	Location	DMs			Cat.	Location	DMs		
		D ₁	D ₂	D ₃			D ₁	D ₂	D ₃
C ₁	L ₁	VG	G	G	C ₄	L ₁	G	G	G
	L ₂	F	F	F		L ₂	VG	F	G
	L ₃	P	P	F		L ₃	P	VP	VP
C ₂	L ₁	G	VG	VG	C ₅	L ₁	G	G	G
	L ₂	VG	P	VP		L ₂	VG	VG	VG
	L ₃	G	G	F		L ₃	F	G	G
C ₃	L ₁	G	VG	G	C ₆	L ₁	G	G	G
	L ₂	VG	VG	VG		L ₂	F	F	VG
	L ₃	G	VG	VP		L ₃	G	F	F

Fuzzy weights of categories and applied Eq. (2) to calculate the weights of categories as follows: $w_1=0.183$; $w_2=0.1948$; $w_3=0.1803$; $w_4=0.133$; $w_5=0.141$; $w_6=0.168$.

For each location, three decision makers use the linguistic variables, as shown in Table 1, to produce fuzzy performance ratings against each criterion in the following way.

6 Experimental Results

Experimental results could be presented after inserting the responses of three locations, six group criteria by three decision makers'. By applying Eq. (4), the aggregated ratings of facility location with respect to the six group criteria can be computed and shown in the following way.

Table 2. Aggregated decision matrix

R_{ij}	Categories					
Location	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆
L ₁	7.5533	8.1067	7.5533	7.0000	7.0000	7.0000
L ₂	5.0000	4.3300	8.6600	6.8867	8.6600	6.2200
L ₃	3.6667	6.3333	5.6633	1.8867	6.3333	5.6667

Table 3. Normalized decision matrix

v_{ij}	Categories					
Location	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆
L ₁	0.7729	0.7263	0.5896	0.7001	0.5464	0.6395
L ₂	0.5117	0.3879	0.6760	0.6887	0.6760	0.5683
L ₃	0.3752	0.5674	0.4421	0.1887	0.4944	0.5177

Normalized decision matrix can be calculated by applying Eq. (5). Determine positive ideal solution, A^+ , and negative ideal solution, A^- in the following way.

$$A^+ = (0.7729, 0.7263, 0.6760, 0.7001, 0.6760, 0.6395)$$

$$A^- = (0.3752, 0.3879, 0.4421, 0.1887, 0.4944, 0.5177)$$

Calculate weighted Euclidean distance of each selection facility location from A^+ and A^- in the following way.

Table 4. The distance measure

Locations	The distance measure	
	d^+	d^-
L_1	0.060935	0.304679
L_2	0.188856	0.227172
L_3	0.292667	0.079219

Obtained relative closeness coefficient and rank the order of the facility location selection of LDC in Novi Sad area: $RCC_1 = 0.8333$, $RCC_2 = 0.5461$, $RCC_3 = 0.2130$.

According to the above shown relative closeness coefficient, the ranking order of the three alternative facility location distribution centers in Novi Sad area is L_1 , L_2 , and L_3 . The best solution and proposed location is L_1 . The result of fuzzy TOPSIS method is the same as the ranking result determined in [14]. This method is capable of revealing the positive and negative preference degree associated with decision makers' alternative and assisting them in making a decision based on group consensus.

7 Conclusion and Future Work

In this paper, the fuzzy TOPSIS method proposed by [5] is employed to select facility distribution centre location in Novi Sad area, when experts disagree. The linguistic terms are represented and used to evaluate the weights of criteria and the rating of each alternative LDC location with respect to various criteria. Minkowski distance function is applied to measure the distance of each alternative from the positive ideal solution and the negative ideal solution. The preference order of available alternative facility locations can be identified according to the relative closeness coefficients.

Experimental results encourage further research. They could be with other official results of the feasibility study of the LDC located in Novi Sad area. In comparison to the official study, which does not show a methodological basis, this research gets the results similar to the empirical results of their solutions in logistics distribution center in an entirely exact way.

Nevertheless, better results can be obtained by introducing genetic algorithm or other optimization and ranking methods to improve precision and to better describe and spot good and bad performance of the considered potential facility locations.

Acknowledgments. The authors acknowledge the support for research project TR 36030, funded by the Ministry of Science and Technological Development of Serbia.

References

1. Bhattacharya, B.K., Sen, S.: On a simple, practical, optimal, output-sensitive randomized planar convex hull algorithm. *Journal of Algorithms* 25(1), 177–193 (1997)
2. Kahraman, C., Ruan, D., Dogan, I.: Fuzzy group decision-making for facility location selection. *Journal Information Sciences* 157(1), 135–153 (2003)
3. Ertugrul, I., Karakasoglu, N.: Performance evaluation of Turkish cement firms with fuzzy analytic hierarchy process and TOPSIS methods. *Expert Systems with Applications* 36(1), 702–715 (2009)
4. Chu, T.C.: Selecting Plant Location via a Fuzzy TOPSIS Approach. *International Journal of Advanced Manufacturing Technology* 20(11), 859–864 (2002)
5. Yong, D.: Plant location selection based on fuzzy TOPSIS. *International Journal of Advanced Manufacturing Technology* 28(7-8), 839–844 (2006)
6. Chou, C.C.: The canonical representation of multiplication operation on triangular fuzzy numbers. *Computers & Mathematics with Applications* 45(10-11), 1601–1610 (2003)
7. Yang, L., Jones, B.F., Yang, S.H.: A fuzzy multi-objective programming for optimization of fire station locations through genetic algorithms. *European Journal of Operational Research* 181, 903–915 (2007)
8. Tzeng, G.H., Huang, J.J.: *Multiple Attribute Decision Making: Methods and Applications*. Chapman and Hall/CRC (2011)
9. Deng, H., Yeh, C.H., Willis, R.J.: Inter-company comparison using modified TOPSIS with objective weights. *Computers & Operations Research* 27(10), 963–973 (2000)
10. Shih, H.S., Shyr, H.J., Lee, E.S.: An extension of TOPSIS for group decision making. *Mathematical and Computer Modelling* 45(7-8), 801–813 (2007)
11. Zanakis, S.H., Solomon, A., Wishart, N., Dublisch, S.: Multi-attribute decision making: A simulation comparison of select methods. *European Journal of Operational Research* 107(3), 507–529 (1998)
12. Sun, C.C., Lin, G.T.R.: Using fuzzy TOPSIS method for evaluating the competitive advantages of shopping websites. *Expert Systems with Applications* 36(9), 11764–11771 (2009)
13. Kahraman, C., Engin, O., Kabak, Ö., Kaya, İ.: Information systems outsourcing decisions using a group decision-making approach. *Engineering Applications of Artificial Intelligence* 22(6), 832–841 (2009)
14. PU "Urban Planning"– Institute of urban planning Novi Sad, A study on the conditions and justification of construction of the freight transport center in Novi Sad", (*JP "Urbanizam - Zavod za Urbanizam Novi Sad": "Studija o uslovima i opravdanosti izgradnje robno transportnog centra u Novom Sadu"*) (Serbian), Novi Sad (2004)

Clinical Careflows Aided by Uncertainty Representation Models

Tiago Oliveira¹, João Neves², Ernesto Barbosa¹, and Paulo Novais¹

¹ CCTC/DI, University of Minho, Braga, Portugal
{toliveira,pjon}@di.uminho.pt,
pg18744@alunos.uminho.pt

² Hospital of Braga, Braga, Portugal
joao.neves@hospitaldebraga.pt

Abstract. Choosing an appropriate support for Clinical Decision Support Systems is a complicated task, and dependent on the domain in which the system will intervene. The development of wide solutions, which are transversal to different clinical specialties, is impaired by the existence of complex decision moments that reflect the uncertainty and imprecision that are often present in these processes. The need for solutions that combine the relational nature of declarative knowledge with other models, capable of handling that uncertainty, is a necessity that current systems may be faced with. Following this line of thought, this work introduces an ontology for the representation of Clinical Practice Guidelines, with a case-study regarding colorectal cancer. It also presents two models, one based on Bayesian Networks, and another one on Artificial Neural Networks, for colorectal cancer prognosis. The objective is to observe how well these two ways of obtaining and representing knowledge are complementary, and how the machine learning models perform, attending to the available information.

Keywords: Clinical Decision Support Systems, Computer-Interpretable Guidelines, Clinical Uncertainty, Machine Learning.

1 Introduction

Currently, the penetration of Clinical Decision Support Systems (CDSSs) in daily healthcare delivery is becoming a reality. There is even evidence that the use of such systems can contribute positively to the improvement of healthcare services, namely in the prevention of medication errors [1], and the improvement of practitioner performance. The main goal of these systems is to help healthcare professionals to make decisions by dealing with clinical data and knowledge. The advent of CDSSs occurred in the middle of the 1960s and the early 1970s. Through the years, CDSSs evolved into three main types [2]: (i) tools for information management (e.g. Electronic Medical Record (EMR) systems); (ii) tools for focusing attention (e.g. alert systems); and (iii) tools for providing patient-specific recommendations. This paper focuses on the last which are tools that provide custom assessments based on sets of patient data.

Different techniques have been used to support the decision making process of CDSSs, they range from mathematical modelling, pattern recognition and statistical analysis of large databases to specific algorithms represented as flowcharts.

This work follows a hybrid approach consisting in specific algorithms combined with models obtained through machine learning processes. The basis for the algorithmic part will be provided by Clinical Practice Guidelines (CPGs) [3], which are systematically developed statements that provide healthcare professionals with instructions regarding specific clinical circumstances. This work proposes an ontology model for the representation of CPG tasks combined with classification models for specific cases where uncertainty is more evident.

The paper is organized as follows. The next section contains a description of the primitives used in the CPG ontology along with a proper case study featuring colorectal cancer (CRC) diagnosis and treatment. In Europe, this is one of the most common forms of cancer (only second to breast cancer) and it affects predominantly the western countries, a group in which Portugal is included [4]. Section three introduces a moment in CRC management that is usually clouded with uncertainty, the prognosis after surgery, as well as a set of models based on Bayesian Networks (BNs) and Artificial Neural Networks (ANNs) for mortality prediction. The last section presents some conclusions about the work done so far and points out to future directions.

2 Clinical Practice Guideline Representation

The approach followed for CPG representation includes an ontology developed in Ontology Web Language (OWL) *McGuinness2004*. OWL-DL (Description Logics) is a highly expressive language comprised of classes (sets of individuals having certain properties), individuals (objects of the domain) and properties (binary relationships between individuals or between individuals and data). The developed ontology is called *CompGuide* and presents a formalisation of guidelines as linked lists of tasks. This approach was based on Computer-Interpretable Guideline (CIG) [6] formalisms that follow the Task Network Model (TNM), representing CPGs as networks, or workflows, of tasks [7]. Such formalisms include the Guideline Interchange Format (GLIF)[8], PROforma [9] and the Standards-based Sharable Active Guideline Environment (SAGE) [10], just to name a few. The following subsections will present the main class primitives and the properties that enable the definition of the order between tasks, as well as temporal and clinical constraints.

2.1 Task Primitives

In *CompGuide* a GPG is represented as an instance of the class *ClinicalPracticeGuideline*. To sanction the nesting of classes, it was considered that all tasks of a guideline are contained in a broader task called *Plan*, to which an individual of *ClinicalPracticeGuideline* is linked through the *hasPlan* object property. Figure 1 shows a graph containing the top classes of *CompGuide*

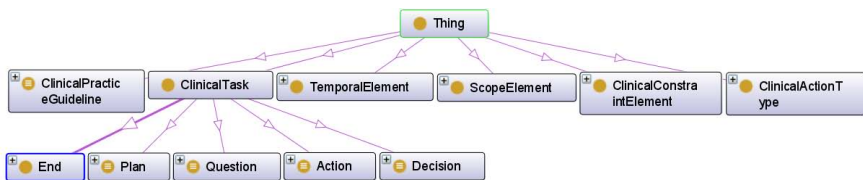


Fig. 1. Representation of the main primitive classes of *CompGuide*

Each guideline has only one *Plan*, and every *Plan* has a variable number of tasks, including other *Plans*. The main classes are subclasses of *ClinicalTask* and consist in *Plan*, *Action*, *Question* and *Decision*. These tasks have properties, or are linked to individuals from other classes in order to express different procedures. Starting with the *Action* class, it is used for steps in the guidelines that must be performed by a healthcare agent, thus encompassing clinical procedures, clinical exams, medication and non-medication recommendations. When some statement concerning a patient has to be asserted, the *Decision* task is used to produce it based on the verification of previously specified conditions and the selection of defined options. The association of conditions to options is done via object properties that link individuals from *Decision* to individuals from *ClinicalConstraintElement*. Feeding this decision process is possible through a *Question* task, which collects all the information necessary for applying a guideline. The individuals that belong to this class have data properties to specify the clinical parameters, and the units under which they should be expressed. Finally, instances of the *End* class are used to signal the end of a careflow.

The definition of a relative order between the tasks is achieved through a set of object properties. A *Plan* is linked to the first of its tasks by the *hasFirstTask* object property, and the task that follows it is connected to the previous by the *nextTask* property. The property ensures the sequential execution of tasks, but leaves out cases where they should be carried out at the same time or alternatively. For these special cases, one uses the *parallelTask* and the *alternativeTask* object properties, respectively.

Figure 2 shows a simplified excerpt from a guideline for diagnosis and management of CRC from the National Comprehensive Cancer Network (NCCN), represented according to *CompGuide*. The main *Plan* of the guideline starts with a *Question* task with the objective of obtaining some specific clinical parameters (e.g. change of bowel habits, occurrence of weight loss and vomiting, among others). Then a *Decision* task is proposed to assess the need for complementary means of diagnosis, based on the answers to the previous task. There are two alternative tasks for the next step, selected according to trigger conditions concerning the result of the *Decision*. If the *Action* task is the one selected, a set of exams is proposed based on which the next *Decision* task will assess the need for CRC surgery. Again, two tasks are shown as an alternative and the selection is carried out the same way as in the previous situation, using trigger conditions according to the possible outcomes of the *Decision*. If the surgery route

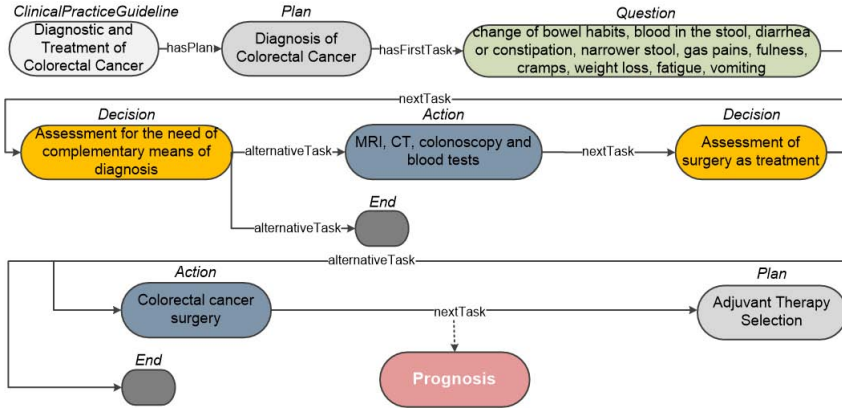


Fig. 2. Excerpt of a guideline for the diagnosis and treatment of colorectal cancer

is followed, then the next step would be a *Plan* for adjuvant therapy selection, i.e., choosing the most appropriate chemotherapy and/or radiotherapy scheme to be applied. In the meantime, there is a prognosis stage whose representation falls outside of the capabilities of the ontology. As such, the integration of the careflow provided by the ontology with a classification model, derived from data, is necessary, as it will be discussed further here-in.

2.2 Temporal Constraints

Besides the relative order by which they are executed, clinical procedures are also bound by temporal constraints, such as duration and cyclic repetitions. In this case, the duration indicates for how long the task should stay active. Hence, to express this, there is the *Duration* class under *TemporalElement*. This class is defined only for *Plans* and *Actions* and each of its individuals has a *decimal* data property, the *DurationValue*, and a *hasTemporalUnit* object property that connects it to an individual from *TemporalUnit*. The available temporal units are *second*, *minute*, *hour*, *day*, *week*, *month* and *year*. In *Loop*, also under *TemporalElement*, one defines the repetition cycles for both *Actions* and *Plans*. An individual from *Loop* has an *integer* data property named *RepetitionValue* where it is possible to specify the number of repetitions of the referred tasks. Another feature is an object property, *hasPeriodicity*, linking to individuals of a class called *Periodicity*. The individuals from this class possess two constructors to define periodic executions of tasks, namely the *PeriodicityValue* data property and the *hasTemporalUnit* object property.

2.3 Clinical Constraints

The execution of tasks depends on the verification of conditions. In a *Decision* task there is a choice between two or more options which are represented by

individuals of the *Option* class under *ClinicalConstraintElement*. The *Option* class is defined by properties that enable the expression of the *Parameter* the option reports to and the value to be asserted to the patient state which might be either a *NumericalValue* or a *QualitativeValue*. For option selection, the definition of conditions is essential. This is done through the *ConditionSet* classes whose instances represent sets of conditions which, in turn, are created as instances of the *Condition* class. The later has appropriate properties to specify the clinical parameter the condition reports to and the value of that parameter that should be checked.

CompGuide also models other types of conditions, namely *TriggerConditions*, *PreConditions* and *Outcomes*, all of them defined under *ClinicalConstraintElement*. A *TriggerCondition* is used to choose the next task in the clinical careflow when they are connected to the previous task by the *alternativeTask* object property. This is accomplished using the *ConditionSet* class in a manner that is similar to *Option*. A *PreCondition* is slightly different in the sense that it represents conditions that must be checked before the application of tasks. Finally, the *Outcome* indicates the expected result after a *Plan*, or an *Action*, that will only be accomplished if their results are met.

3 Management of Clinical Uncertainty

Uncertainty may be defined as something that is not certain and transmits doubts, being an important concept in the medical domain. Indeed, a symptom may be viewed as an uncertain indication of a disease, since it may occur or not together with a certain health condition [11]. The prediction of the expected outcome of a treatment process is one of the responsibilities of healthcare professionals, and is also the moment of the clinical encounter in which uncertainty affects more the decisions. Prognosis may be defined as the prediction of the future course of a disease process that depends on the patient's health history. In this case, the fundamental objective is to predict mortality after 30 days of CRC surgery. This is a critical aspect for surgeons because the death of a patient during this period is considered their direct responsibility. Moreover, physicians could use such a prediction model to identify patients at higher risk, thus acting as a useful complement to CIGs deployed in a CDSS.

3.1 The Case of Colorectal Cancer Prognosis

CRC develops in the cells lining the colon when they suffer mutations that cause their uncontrollable growth. They begin to invade healthy tissues, yielding malignant tumours and may also spread to other parts of the body by entering the bloodstream or the lymphatic system [18].

The existing approaches to CRC prognosis include the following scoring systems: the Physiological and Operative Severity Score for the Enumeration of Mortality and Morbidity (POSSUM), the Portsmouth POSSUM (P-POSSUM) and the Colorectal POSSUM (Cr-POSSUM)[13]. POSSUM is the oldest model

and, being designed for general surgery, it is the one that presents the worst performance when predicting CRC mortality. The P-POSSUM is an evolution of the previous in order to overcome the problem of overpredicting mortality in low risk patients. As for the Cr-POSSUM, it is the latest model, dating from 2004, and it has better calibration and discrimination than the existing POSSUM and P-POSSUM scoring systems. A significant disadvantage of the POSSUM, P-POSSUM and Cr-POSSUM models is that they have not been extensively adopted because their performance is poor in populations different from the ones that yielded the sample on which their development was based [13].

There are many variables that influence CRC prognosis, this being one of the reasons why this process is so problematic. The other reason is the interactions between the variables and the effect they have on the outcome, which are not entirely known and, as such, are difficult to deal with, even when one is rigorously following a CPG. For these situations, other models are necessary for the completion of guidelines and to build a complete solution for the management of diseases and treatments. From the literature it was possible to isolate a set of variables considered important for CRC prognosis and group them under two classes [12,13]: physiological factors and operative severity factors.

The physiological factors describe the physical condition of a patient, thus including [12,13]: age, sex, cardiac signal, respiratory signal, ElectroCardioGram (ECG) findings, systolic blood pressure, diastolic blood pressure, cardiac frequency, levels of substances in the blood (e.g. haemoglobin, leukocytes, sodium and potassium), urea levels, Dukes cancer classification and the American Society of Anaesthesiologists (ASA) physical status classification.

On the other hand, the operative severity factors include elements related with the surgery that affect the patient's recovery [12,13]. This class consists of: pathology type, surgical urgency, surgical approach, operative severity (as classified by the surgeon), total blood loss, contamination of the peritoneal cavity, type of CRC procedure and cancer resection status (i.e., if the tumour is technically removable or not).

These were the factors used for the construction of the models presented in the next section. They were used as inputs for the models to predict the outcome expressed as 30-day mortality after surgery.

3.2 Developed Models

This work configures a case of supervised learning, since one is trying to infer a function and make a generalization based on labelled data. The data set used corresponds to a sample of 230 patients that received surgical treatment for CRC at the Hospital of Braga. The attributes in the data set are the factors presented in the previous section, regarded as inputs for a classifier. The outcome of the classifier was considered to be the mortality within 30 days after surgery, with the possible values *yes* or *no*.

There is a number of machine learning models for supervised learning, according to the way they represent information. For this work, Bayesian Networks

(BNs) and Artificial Neural Networks (ANNs) were considered as potential solutions for modelling CRC prognosis. A Naïve Bayes classifier is a probabilistic model that uses Bayes rule and has a graphical representation in the form of a directed graph, it is characterized by the assumption that all attributes (inputs) are independent [14]. Although the independence assumption is a simplistic one for real life, this type of classifier usually performs well in actual data sets. As for ANNs, they are a mathematical approach inspired in biological neuron networks that consist in an interconnected group of artificial neurons, each one having a specific activation function. Arguably the most well known form of ANN is the Multi-Layer Perceptron which is an ANN that trains using backpropagation and consists in multiple layers containing neurons, namely an input layer connected to a variable number of hidden layers, which in turn are connected to an output layer.

Using the Classify tab in the Weka Explorer interface, a *NaiveBayes* and a *MultilayerPerceptron* classifiers were obtained. For testing purposes, 5-fold cross-validation was performed, producing the results (expressed as mean values) shown in Table 1. A *NominaltoBinary* filter was applied to the data set but the remaining parameters used for learning the *MultilayerPerceptron* were set to default since they were the ones that yielded a better result in terms of the performance measures used in this work. Changes in the learning rate, the number of hidden layers and momentum usually resulted in a worst performance.

Table 1. 5-fold cross-validation results for the CRC prognosis classifiers

<i>Classifier</i>	<i>Kappa statistic</i>	<i>Mean abs. error</i>	<i>Precision</i>	<i>Recall</i>	<i>F-measure</i>
NaiveBayes	0.192	0.0715	0.927	0.939	0.932
MultilayerPerceptron	-0.0452	0.093	0.904	0.913	0.908

The *Kappa statistic* measures the agreement between two raters, in this case, between each classifier and the true classes in the data set [15,16]. This measure removes the probability of chance agreement and if a classifier has a value higher than 0, as it is the case of the *NaiveBayes*, it means that said classifier is performing better than chance. On the other hand, a value inferior to 0 means that agreement occurs less than it was predicted by chance. This happens with the *MultilayerPerceptron*, revealing a poor correspondence with reality. In turn, the *mean absolute error* summarises how close forecasts are to eventual outcomes [17] and, in this parameter, the *NaiveBayes* is associated with a lower error than its counterpart, thus indicating a minor deviation from the real values of the labels, though not by much.

Precision and *recall* are measures that are usually used in pattern recognition to assess model performance [14]. The first corresponds to the fraction of instances classified as positive that are true positives, while the second represents the fraction of positives that were correctly classified. The *F-measure* is the harmonic mean of the previous two, a combined score [14]. The values of Table 1 are the weighted averages of these three measures, in which the *NaiveBayes* shows

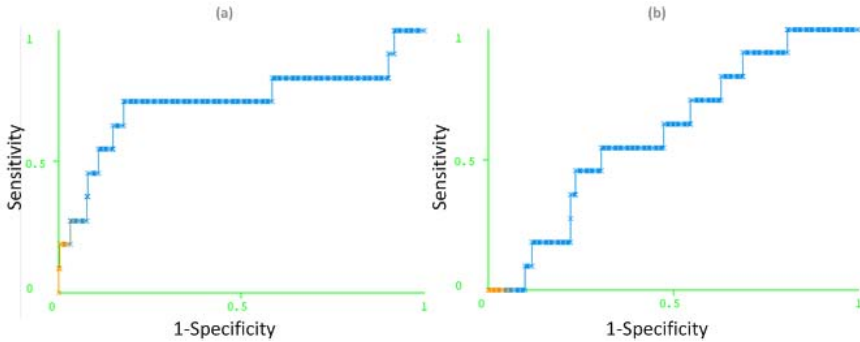


Fig. 3. Receiver operating characteristic for (a) the *NaiveBayes* and (b) the *MultilayerPerceptron* classifiers, regarding the class *yes* represents a death during the 30 day period

an overall better performance. However, these values may be misleading once the occurrence of an actual death is a rare phenomenon, which translates into the classification classes not being approximately equally represented. In fact, the values for *precision*, *recall* and *F-measure* for class *yes* were abnormally low, whilst for class *no* they were significantly higher, which results in a high weighted average. This class imbalance results from the difficulty to obtain data of deceased patients, evident in the sample studied for this work where there are only 11 cases of death out of 230 instances. Only about 100 patients are submitted to CRC surgery in the Hospital of Braga and most medical files are not computerized, so their consultation is a slow process.

The results of the Receiver Operating Characteristics (ROC) of Figure 3 are in consonance with the ones already shown. A ROC curve is a graphical plot of *sensitivity*, also called true positive rate, against *1-specificity*, or probability of false alarm, that evaluates the performance of a binary classifier. The desired result is to have low values of *1-specificity* for high values of sensitivity, i.e., the biggest possible are under the curve. As the graphics show, the *NaiveBayes* classifier is the one with a bigger area under curve, with a value of 0.795, against 0.668 of the *MultilayerPerceptron*.

4 Conclusions and Future Work

This work suggests an alternative to the purely rule-based methods for clinical decision, addressing the limitations of explicit knowledge. This enables the system to tackle problems such as high complexity situations and uncertainty.

The *CompGuide* ontology deals with the definition of clinical tasks, their ordering and scheduling, in a care flow with different plans. Care flow management systems with an underlying ontology allow an advanced reasoning and the sharing of a standard representation. However, the representation of clinical information requires an inherent flexibility, given the variability of decision making

processes that one may find in different medical domains. CRC prognosis is one of such cases, where healthcare professionals require more powerful tools than simple CPG algorithms. This calls for the inclusion of models capable of representing complex and uncertain information in the procedural logics of the CIG execution engine.

Two classifiers were produced to forecast the outcome of the prognosis after CRC surgery. The *NaiveBayes* classifier was the one that showed a better performance. Being a graphical model, it is also better at delivering information to healthcare professionals. The belief network enables the users to selectively condition each entry variable and verify its impact on the outcome variable, in the form of a probability adjustment. This is more advantageous over the opaqueness of ANNs and thus the *MultilayerPerceptron*, where it is possible to view the system in terms of inputs and outputs, but not its internal workings. Healthcare professionals consider the inference process as equally valuable as the outcome. As so, it may be concluded that the Bayesian model is the best choice for integration with the care flow modelled by the ontology. Being so, it is also noticeable that the model needs refinement by extracting the most relevant features in order to make better generalizations and achieve better performances. Moreover, some pre-processing with techniques to adjust imbalanced data sets, such as the (Synthetic Minority Oversampling TEchnique) SMOTE [19], is needed in order to see if they increase the *NaiveBayes* performance.

The next steps include the improvement of the current classifiers as well as a comparison with the Cr-POSSUM score. The development of other models of the same type for other key moments of the guideline depicted in Figure 2, such as the prediction of patient response to adjuvant therapy, is also a research line to be followed. The goal is to build a general solution capable of providing personalized recommendations.

Acknowledgements. This work is funded by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project PEst-OE/EEI/UI0752/2011". The work of Tiago Oliveira is supported by a doctoral grant by FCT (SFRH/BD/85291/2012).

References

1. Kaushal, R., Shojania, K.G., Bates, D.W.: Effects of computerized physician order entry and clinical decision support systems on medication safety: a systematic review. *Archives of Internal Medicine* 163(12), 1409–1416 (2003)
2. Musen, M.A., Shahar, Y., Shortliffe, E.H.: Clinical decision-support systems. *Biomedical Informatics*, 698–736 (2006)
3. Rosenbrand, K., Croonenborg, J., Wittenberg, J.: Guideline Development. In: Teije, A., Miksch, S., Lucas, P. (eds.) *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends*, pp. 3–22 (2008)
4. Ferlay, J., Autier, P., Boniol, M., Heanue, M., Colombet, M., Boyle, P.: Estimates of the cancer incidence and mortality in Europe in 2006. *Annals of Oncology: Official Journal of the European Society for Medical Oncology* 18(3), 581–592 (2007)

5. McGuinness, D.L., Van Harmelen, F.: OWL Web Ontology Language Overview. W3C Recommendation 10, 1–19 (2004)
6. Isern, D., Moreno, A.: Computer-based execution of clinical guidelines: a review. *International Journal of Medical Informatics* 77(12), 787–808 (2008)
7. Oliveira, T., Novais, P., Neves, J.: Development and implementation of clinical guidelines: An artificial intelligence perspective. *Artificial Intelligence Review* (2013), doi: 10.1007/s10462-013-9402-2
8. Ohno-Machado, L., et al.: The guideline interchange format. *Journal of the American Medical Informatics Association* 5(4), 357 (1998)
9. Vier, E., Fox, J., Johns, N., Lyons, C., Rahmzadeh, A., Wilson, P.: PROforma: systems. *Computer Methods and Programs in Biomedicine* 2607(97) (1997)
10. Tu, S., et al.: The SAGE Guideline Model: achievements and overview. *Journal of the American Medical Informatics Association* 14(5), 589–598 (2007)
11. Straszeka, E.: Combining uncertainty and imprecision in models of medical diagnosis. *Information Sciences* 176, 3026–3059 (2006)
12. Horzic, M., Kopljar, M., Cupurdija, K., Bielen, D.V., Vergles, D., Lackovic, Z.: Comparison of P-POSSUM and Cr-POSSUM scores in patients undergoing colorectal cancer resection. *Archives of Surgery* 142(11), 1043–1048 (2007)
13. Senagore, A.J., Warmuth, A.J., Delaney, C.P., Tekkis, P.P., Fazio, V.W.: POSSUM, p-POSSUM, and Cr-POSSUM: implementation issues in a United States health care system for prediction of outcome for colon cancer resection. *Diseases of the Colon and Rectum* 47(9), 1435–1441 (2004)
14. Witten, I.H., Frank, E., Hall, M.: *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann (2011)
15. Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20(1), 37–46 (1960)
16. Lantz, C.A., Nebenzahl, E.: Behavior and interpretation of the κ statistic: Resolution of the two paradoxes. *Journal of Clinical Epidemiology* 49(4), 431–434 (1996)
17. Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research* 30(1), 79–82 (2005)
18. Winawer, S., Fletcher, R., Rex, D., Bond, J., Burt, R., Ferrucci, J., Ganiats, T., Levin, T., Woolf, S., Johnson, D., Kirk, L., Litin, S., Simmang, C.: Colorectal cancer screening and surveillance: Clinical guidelines and rationale-Update based on new evidence. *Gastroenterology* 124(2), 544–560 (2003)
19. Chawla, N.V.: Data mining for imbalanced datasets: An overview. In: *Data Mining and Knowledge Discovery Handbook*, pp. 853–867. Springer (2005)

A Hybrid Approach for the Verification of Integrity Constraints in Clinical Practice Guidelines

Marco Iannaccone, Massimo Esposito, and Giuseppe De Pietro

National Research Council of Italy - Institute for High Performance Computing and Networking (ICAR), Naples, Italy
{iannaccone.m, esposito.m, depietro.g}@na.icar.cnr.it

Abstract. In the last decade, clinical practice guidelines are increasingly implemented in decision support systems able to promote their better integration into the clinical workflow. Despite the attempts involved to detect malformed, incomplete, or even inconsistent implementations of computerized guidelines, none of these solutions is concerned with directly embedding the theoretic semantics of a formal language as the basis of a guideline formalism in order to easily and directly support its verification. In such a direction, this paper proposes a formal framework which has been seamlessly embedded into a standards-based verifiable guideline model, named GLM-CDS (GuideLine Model for Clinical Decision Support). Such a framework hybridizes the theoretic semantics of ontology and rule languages to codify clinical knowledge in the form of a process-like model and, contextually, specify a set of integrity constraints to help to detect violations, errors and/or missing information. Its strong point relies on the capability of automatically verifying guidelines and, thus, supporting developers without the necessary technical background to construct them in a well-formed form. As a proof of concept, an actual guideline for Advanced Breast Cancer has been used to highlight some malformed implementations violating integrity constraints defined in GLM-CDS.

Keywords: Clinical Practice Guidelines, Decision Support Systems, Knowledge Verification, Ontology, Rules.

1 Introduction

In the last years, Evidence-Based Medicine has given a major impetus to the development of Clinical Practice Guidelines (hereafter, CPGs) as a means to promote standards of medical care and avoid substandard practices or outcomes. Even if CPGs are conceived to assess and improve the quality of care for patients, minimizing the variance in the care and cutting down costs of medical services [1,2], the general practitioners' behaviour is not effectively improved by disseminating CPGs in the form of educational paper-based documents [3]. In order to address this issue, CPGs are increasingly implemented in computer-based decision support systems (hereafter, DSSs) able to promote a better integration into the clinical workflow and provide point-of-care patient-specific recommendations [4]. A frequently reported factor that

complicates the development of CPGs in computer-based DSSs is the additional effort, beyond conceiving the CPGs, needed to structure them in the form of computer-interpretable guidelines (hereafter, CIGs) due to the inconsistency and poverty of the methodological rigor used to computerize guideline knowledge [5]. Indeed, despite the fact that CPGs are issued by experts' panels, their encoding can produce CIGs that might be malformed, incomplete, or even inconsistent [6].

To date, the verification capabilities offered by existing guideline languages are usually rather limited and, only recently, this limitation has led to proposals based on formal verification techniques [5]. Unfortunately, none of these solutions is fully integrated into a guideline formalism, since all they require a preliminary translation of a CIG into a formal specification language for enabling the automatic verification [5].

Recently, we proposed a standards-based process-flow-like model, named GLM-CDS (hereafter, GuideLine Model for Clinical Decision Support) [7]. It synthesizes prior work done in the guideline modeling community and directly integrates the model standardized as Health Level 7 Virtual Medical Record for Clinical Decision Support (hereafter, HL7 vMR)[8], to explicitly define concepts and data that are used in a CIG, so that they can be seamlessly mapped to electronic health record entries.

In order to enable the verification of CIGs and face the drawback of the aforementioned solutions, this paper proposes a formal framework, seamlessly embedded into GLM-CDS, which hybridizes the theoretic semantics of ontology and rule languages to codify clinical knowledge in the form of a process-like model and, contextually, specify a set of integrity constraints, i.e. axioms/formulae, built on the top of it in order to help to detect violations, errors and/or missing information. The solution here proposed is particularly relevant for the design and development of a CIG, where formulation of knowledge and construction of a formal process-like model are intermingled [9]. Indeed, the automatic verification of CIGs could be potentially more beneficial for supporting guideline developers without the necessary technical background to construct them in a well-formed form.

The rest of the paper is organised as follows. Section 2 outlines an overview of the state-of-the-art solutions existing in literature. In Section 3, the formulation of the proposed approach is described. Section 4 depicts an example application to an existing guideline in order to highlight a set of malformed possible implementations which violate ICs defined in GLM-CDS. Finally, Section 5 concludes the work.

2 Related Work

To date, the effort in defining new models for encoding CPGs has not been coupled by a parallel effort in embedding systematic approaches for verifying the well-formedness of the resulting CIGs. In the last two decades, some solutions, based on knowledge-based techniques, have been proposed to verify CIGs, but their application was strictly limited to CIGs conceived as sets of condition-action pairs [10-13]. In particular, the approaches described in [10-12] rely on logical analysis and decision table techniques to verify CIGs in terms of completeness, unambiguousness and coherency. Moreover, a three-step method is described in [13] to verify conditions

within a CIG conceived as a hierarchy of plans, in order to detect violations of certain properties formulated within an admissible CIG's condition set.

More recently, many approaches have designed and developed as process-flow-like models for representing CPGs with a different coverage and particularities [14], but offering rather limited verification capabilities [24]. In order to face this limitation, some proposals have appeared, based on formal verification techniques, such as theorem proving or model checking [6], [15-20]. In detail, formal methods have been used in [15] in the form of an interactive verification to improve the quality of guidelines, with the focus to the management of jaundice in newborns. Similarly, in the European project named *Protocure*, a medical protocol, modelled as a hierarchical plan, is mapped to the formal specification of an interactive theorem prover for higher order logic [16,17]. On the other hand, in the European project named *Protocure II*, model checking techniques [18,19] have been explored to support the fully automatic verification in contrast to interactive verification based on theorem proving. Similarly, the approach proposed in [6] is based on the integration of a computerized guideline management system with a model checker to verify CIGs. A framework based on a Model-Driven Approach is described in [20], aimed at authoring and verifying CPGs. Here, CPGs are encoded using a UML tool and the generated formal CIGs are given as input to a model checker for detecting semantic errors and inconsistencies.

A drawback common to all these proposals based on formal verification techniques is represented by the lack of a full integration with a guideline language. Indeed, a CIG, encoded according to a predefined guideline model, has to be first translated into a formal specification language, and, then, can be automatically checked [5]. This loosely coupled approach implies that a guideline model cannot be directly and rigorously described via the well-defined theoretic semantics of a formal language, but only by specifying, a posteriori, some properties a modeled CIG should hold. The solution here proposed has been conceived to face this issue in order to easily and directly support guideline verification, as described in the following sections.

3 The Hybrid Approach for a Verifiable Guideline Model

3.1 The GuideLine Model for Clinical Decision Support

The GLM-CDS synthesizes prior work done in the guideline modeling community and integrates the Domain Analysis Model, Release 1 of the HL7 vMR standard (hereafter, HL7 vMR-DAM) issued by HL7 CDS-WG, by focusing on issues pertaining the clinical decision support.

It consists of a control-flow part, which is based on a formal Task-Network Model (hereafter, TNM) for codifying CPGs in terms of structured tasks connected with transition dependencies between them from an initial state of the patient. Its information model is coded in terms of guideline knowledge coherently with the HL7 vMR-DAM. Existing standard terminological resources, such as Logical Observation Identifiers Names and Codes (LOINC) [21] and Systematized Nomenclature of MEDicine (SNOMED)[22] are used for its population with appropriate semantic

content. Data types used in GLM-CDS resemble the ones defined in the HL7 vMR DAM, which gives a simplified/constrained version of ISO 21090 data types, based on the abstract HL7 version 3 data types specification, rel. 2 [23].

Each guideline in GLM-CDS is described in terms of four main elements: *Guideline*, *EntryPoint*, *ExitPoint*, and *Task*. In more detail, each (sub)guideline is represented by the TNM element named *Guideline*, whereas *EntryPoint* and *ExitPoint* represent start and end points of the TNM. The element *Task* is generic and specialized into the following sub-elements: *Action*, *Decision*, *Condition*, *Split* and *Merge*.

Action models a high-level action to be performed, which is specialized into the sub-elements *Observation*, *Supply*, *Encounter*, *Procedure* and *SubstanceAdministration*. *Observation* is used to determine a measurement, a laboratory test or a user input value. *Supply* is aimed at providing some clinical material or equipment to a patient. *Encounter* is applied to request an appointment between a patient and healthcare participants for assessing his health status. *Procedure* models an action whose outcome is the alteration of the patient's physical condition. Finally, *SubstanceAdministration* allows giving a substance to a patient for enabling a clinical effect. Each action encapsulates a list of one or more elementary and repeatable action items, namely *ObservationItem*, *SupplyItem*, *EncounterItem*, *ProcedureItem* and *SubstanceAdministrationItem*, expressed in terms of the HL7 vMR-DAM.

Decision models decision criteria for directing the control-flow from a point into the TNM to various alternatives. *Condition* is defined as an observable state of the patient that persists over time and tends to require intervention or management. It allows synchronizing the management of a patient with the corresponding guideline or parts of it. Finally, *Split* and *Merge* enable to direct the guideline flow to multiple parallel tasks. In particular, *Split* allows branching to multiple tasks, whereas *Merge* allows synchronizing parallel tasks by making them converging into a single point.

3.2 The Hybrid Framework for Computerizing GLM-CDS

The computerization of GLM-CDS has been realized via a formal framework which hybridizes the theoretic semantics of ontology and rule languages relying on Description Logics (hereafter, DLs) and Logic Programming (hereafter, LP).

In detail, DLs are decidable fragments of first-order logic with a syntax having, as basic building blocks, the notion of concepts (unary predicates, classes), individuals (instances of concepts), abstract roles (binary predicates between concepts) and concrete roles (binary predicates between concepts and data values). In contrast with first-order logic, they allow for decidable reasoning by means of a set of axioms, whose semantics is given by first-order interpretations and, thus, obeys the Open World Assumption (hereafter, OWA), i.e., a statement cannot be inferred to be false on the basis of failures to prove it. Moreover, it does not adopt the Unique Name Assumption (hereafter, UNA), i.e., two different names may refer to the same object.

However, since DLs do not enable a more extensive definition of roles through axioms, they have been here hybridized with LP according to OWA. In particular, rules, which are widely considered in literature as a syntactic and semantic extension

to DLs, have been used as a new kind of axiom to define abstract roles as well as arithmetic relationships between data values assumed by concrete roles.

In the context of pure conceptual reasoning on GLM-CDS, such an OWA is suitable and desirable. However, when the conformity of some instance data to an ontology has to be evaluated, all the DL axioms plus rules should behave more like ICs. In other words, the ontology should be treated as a schema language for the instance data, thus effectively implementing a Closed World Assumption (hereafter, CWA), i.e., any statement that is not specified is assumed to be false. For instance, according to the IC semantics, domain/range constraints or cardinality restrictions for roles should be treated as checks under CWA and UNA rather than inference rules.

However, even if adopting a global CWA can enable the verification of CIGs against GLM-CDS, encoded in the form of ontology enriched with rules, a drawback of this solution is that it requires an “all-or-nothing” choice. In other words, all the information about the domain contained in GLM-CDS is assumed to be complete and, thus, axioms are always considered as integrity constraints to be checked, instead of deductive rules to be inferred, while evaluating instance data.

In order to address this issue and provide more flexibility, the formal framework here proposed has been thought to preserve the semantics of the hybrid knowledge base, which implements GLM-CDS by combining DLs and LP under OWA, and, plus, to support a local CWA by formulating ICs as special rules to be checked against the hybrid knowledge base. More formally, such a solution has been achieved via the methodology detailed as follows.

Let us consider a generic ontology language \mathcal{O} belonging to DL and a rule language \mathcal{R} belonging to a subset of LP, i.e. Horn Clause Logic (hereafter, HCL)

The language \mathcal{O} is made of non-empty and disjoint sets of predicates \mathcal{O}_p , containing concepts and roles, and individual constants \mathcal{O}_i . A collection of terminology, role and assertion axioms, denoted as \mathcal{O}_{ax} , can be expressed over \mathcal{O}_p and \mathcal{O}_i . Terminology, role and assertion axioms are typically denoted as *TBox*, *RBox* and *ABox*.

The language \mathcal{R} is made of non-empty and pair-wise disjoint sets of variables \mathcal{R}_v , individual constants \mathcal{R}_i and rule predicates \mathcal{R}_p . By exploiting such alphabets of \mathcal{R} an *atom* is defined as an expression in the form $a(T)$, where $a \in \mathcal{R}_p$ and T is a tuple of variables $v \in \mathcal{R}_v$ or constants $i \in \mathcal{R}_i$. A *rule* is specified as *Horn clause*, i.e., a disjunction of positive or negative atoms in the form:

$$r_l(X_l) \vee \neg a_l(Y_l) \vee \dots \vee \neg a_k(Y_k) \quad (1)$$

where $r_l, a_l, \dots, a_k \in \mathcal{R}_p$ and $X_l, Y_l, \dots, Y_k \in \mathcal{R}_v$ or \mathcal{R}_i . For extending expressiveness of DL axioms, *definite* Horn clauses, i.e., clauses containing only one positive atom, are here considered, which can be written as follows:

$$r_l(X_l) \leftarrow a_l(Y_l) \wedge \dots \wedge a_k(Y_k) \quad (2)$$

where $r_l(X_l)$ is named *head*, whereas $p_l(Y_l) \dots p_k(Y_k)$ (with $k \geq 0$), are called *body*.

With these premises, the *hybrid Knowledge Base* $K = (\{O\}, \{R\})$ is defined as a finite set of ontology axioms $\{O\}$ belonging to \mathcal{O}_{ax} in the ontology language \mathcal{O} , and a

finite set of deductive *safe* rules $\{R\}$ over \mathcal{O} and \mathcal{R} , which hold the conditions that $\mathcal{R}_p \supseteq \mathcal{O}_p$, $\mathcal{O}_i \equiv \mathcal{R}_i$, and each variable of its head appears also in the body.

Moreover, a finite set of *integrity constraints* $\{IC\}$ is defined over the *hybrid Knowledge Base* K in the form of a *negative* Horn clause, i.e. a clause containing no positive atom, which holds the conditions that $\mathcal{R}_p \supseteq \mathcal{O}_p$ and $\mathcal{O}_i \equiv \mathcal{R}_i$.

For the sake of uniformity, each IC is associated with a special predicate IC_i , so as to have the same form of deductive rules, as formulated in (3):

$$IC_i \leftarrow a_l(Y_l) \wedge \dots \wedge a_k(Y_k) \quad (3)$$

and, thus, an IC encoded as a rule is violated when its special predicate is entailed by the *hybrid Knowledge Base* K . According to this formal asset, K is still interpreted under OWA, whereas the set of ICs $\{IC\}$ is interpreted under CWA.

3.3 The Hybrid Knowledge Base and the Integrity Constraints

The above-described hybrid methodology has been applied to formally computerize the GLM-CDS and produce both a hybrid knowledge base and a set of ICs on the top of it. In detail, all the basic elements forming a TNM in GLM-CDS have been computerized as ontology concepts whereas all their specializations as subsumed ones. All the elements not appearing into the TNM but composing the information model of GLM-CDS, such as action items or decision models, have been encoded as ontology concepts, as well. All the concept definitions are grouped to form the TBox.

Moreover, a number of abstract and concrete roles has been defined for expressing relationships between concepts or specifying data values a concept can assume. In the following, concepts are indicated with the first letter capitalized, whereas abstract and concrete roles are entirely written in lowercase. For instance, *label* and *status* are two concrete roles which associate a *Guideline* or a *Task* with a textual identifier and the possible states they can assume, respectively. On the other hand, examples of abstract roles are *entryPoint*, *exitPoint*, *subguideline* and *task*, which enable to associate a *Guideline* with a collection of tasks or sub-guidelines linked together in a TNM. Other examples of abstract roles, conceived with a topological semantics, are *connectedTo/indirectConnectedTo* and their inverse roles *connectedFrom/indirectConnectedFrom*. They are specified for all the concepts representing the nodes of a TNM to enable a direct/indirect connection among them, i.e. without or with other intermediary nodes between them.

Finally, the RBox has been enriched with some deductive rules for defining more complex axioms. For instance, in order to define the role *indirectConnectedTo*, the rule $indirectConnectedTo(x,z) \leftarrow connectedTo(x,y) \wedge connectedTo(y,z)$ has been added to the RBox. For major details about all the concepts, roles and the relative axioms/rules defined in GLM-CDS, please refer to [7].

On the top of these TBox and RBox forming the hybrid knowledge base, some of the ICs formulated in GLM-CDS are described in natural language in Table 1. This set of ICs is mainly aimed at granting a well-formed TNM encoding a CIG.

Table 1. ICs defined in GLM-CDS, formulated in natural language

IC	Description
1	Each <i>Guideline</i> must be made by exactly one <i>EntryPoint</i> , one <i>ExitPoint</i> and at least one <i>Task</i>
2	Each <i>EntryPoint/ExitPoint</i> is not admissible to have a direct connection from/to any TNM element.
3	Each <i>Task</i> must be directly connected at least to and from another TNM element and at most to one <i>ExitPoint</i> and from one <i>EntryPoint</i> .
4	<i>Split/Merge</i> is admissible to have a direct connection only to/from a <i>Task</i> .
5	Each set of parallel tasks must start and terminate with exactly one <i>Split</i> and <i>Merge</i> .

These ICs have been next formalized in the form of rules with a special predicate in their heads as indicated in Table 2. The notation adopted in Table 2 specifies concepts and roles as unary and binary predicates, respectively, where $Concept(x)$ is used to test if a variable x is an instance of *Concept*, whereas $role(x)$ is adopted to test if a variable x is linked to a variable y by means *role*. Moreover, the rule binary predicate $notEqual(x,y)$ is used to test if a variable x is different from a variable y . For each IC, a set of one or more rules is produced, whose single entailment against the hybrid knowledge base is sufficient to assert that such an IC is violated.

Table 2. ICs defined in GLM-CDS, formulated as rules

IC	Rules
1	$IC1 \leftarrow Guideline(x) \wedge entryPoint(x,y) \wedge EntryPoint(y) \wedge entryPoint(x,z) \wedge EntryPoint(z) \wedge notEqual(y,z)$ $IC1 \leftarrow Guideline(x) \wedge exitPoint(x,y) \wedge ExitPoint(y) \wedge exitPoint(x,z) \wedge ExitPoint(z) \wedge notEqual(y,z)$ $IC1 \leftarrow Guideline(x) \wedge not\ task(x,y)$
2	$IC2 \leftarrow ExitPoint(x) \wedge connectedTo(x,y)$ $IC2 \leftarrow EntryPoint(x) \wedge connectedFrom(x,y)$
3	$IC3 \leftarrow Task(x) \wedge not\ connectedTo(x,y)$ $IC3 \leftarrow Task(x) \wedge not\ connectedFrom(x,y)$ $IC3 \leftarrow Task(x) \wedge connectedTo(x,y) \wedge ExitPoint(y) \wedge connectedTo(x,z) \wedge ExitPoint(z) \wedge notEqual(y,z)$ $IC3 \leftarrow Task(x) \wedge connectedFrom(x,y) \wedge EntryPoint(y) \wedge connectedFrom(x,z) \wedge EntryPoint(z) \wedge notEqual(y,z)$
4	$IC4 \leftarrow Split(x) \wedge not\ connectedTo(x,y) \wedge Task(x)$ $IC4 \leftarrow Merge(x) \wedge not\ connectedFrom(x,y) \wedge Task(x)$
5	$IC5 \leftarrow Split(x) \wedge indirectConnectedTo(x,y) \wedge Merge(y) \wedge indirectConnectedTo(x,z) \wedge Merge(z) \wedge notEqual(y,z)$ $IC5 \leftarrow Merge(x) \wedge indirectConnectedFrom(x,y) \wedge Split(y) \wedge indirectConnectedFrom(x,z) \wedge Split(z) \wedge notEqual(y,z)$

4 An Example Application: A CIG for Advanced Breast Cancer

This section reports, as an example, the application of GLM-CDS to the guideline for the “Advanced Breast Cancer”, issued by the National Institute for Health and Care Excellence (hereafter, NICE) in February 2009. Such a guideline has been partially encoded according to the GLM-CDS with reference to the imaging assessment of the disease by obtaining the well-formed CIG reported in Figure 1. For the sake of clarity,

this CIG is represented on multiple levels of abstraction, i.e. sub-guidelines are used, for instance, to group recommendations about treatment. However, due to space limitations, their content is not further detailed.

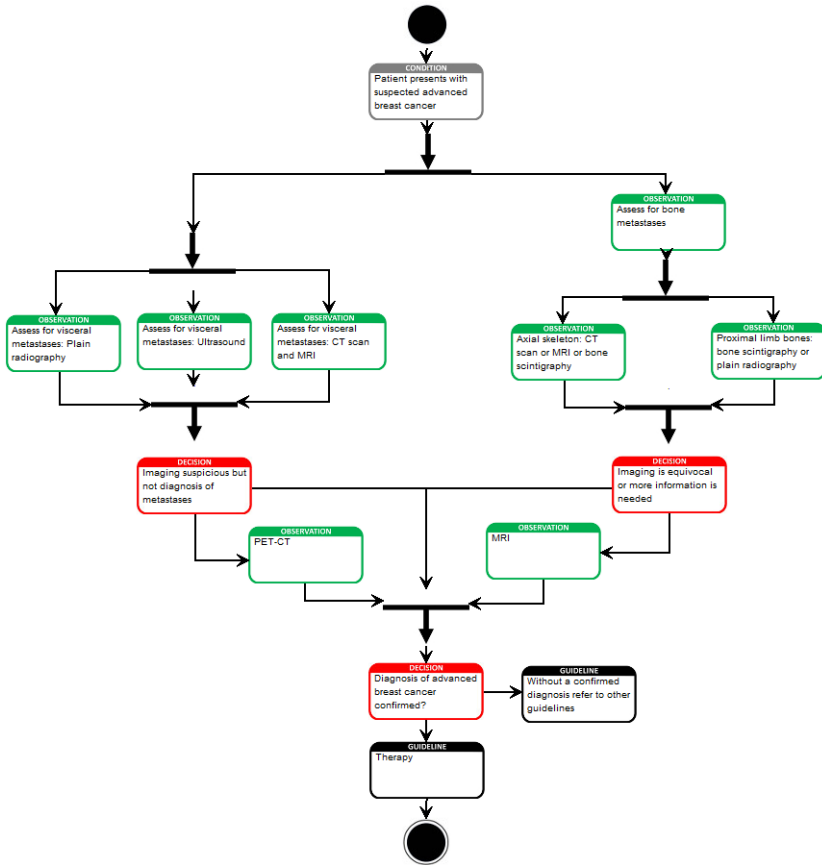


Fig. 1. The NICE guideline for Advanced Breast Cancer encoded in GLM-CDS

In order to highlight how the ICs formulated in the previous section operate on a malformed CIG encoded according to GLM-CDS, some examples have been reported in Figure 2, built starting from the NICE guideline for Advanced Breast Cancer.

In detail, the CIG shown in Figure 2a) is made of only an *EntryPoint* and an *ExitPoint* and, thus, since the IC 1 is violated, it is detected as malformed. The CIG reported in Figure 2b) violates the IC 2, since the *EntryPoint* has a connection from a TNM element and the *ExitPoint* has a connection to a TNM element. Both the ICs 1 and 3 are not verified in the CIG depicted in Figure 2c), since that CIG contains two *ExitPoints*, and, plus, the *Condition* is simultaneously connected to two *ExitPoints*. The IC 3 is not hold by the CIG depicted in Figure 2d), since the *Condition* is not connected to any other TNM element. The CIG scratched in Figure 2e) is not well-formed since *Split* and *Merge* are not connected to other tasks but are directly linked

between them, so violating the IC 4. Finally, the IC 5 violates the CIG depicted in Figure 2f) since the *Split* generates three parallel sequences of tasks, but only two of these sequences are closed into a *Merge*, whereas the third sequence is connected to another TNM element, i.e. the next *Decision*.

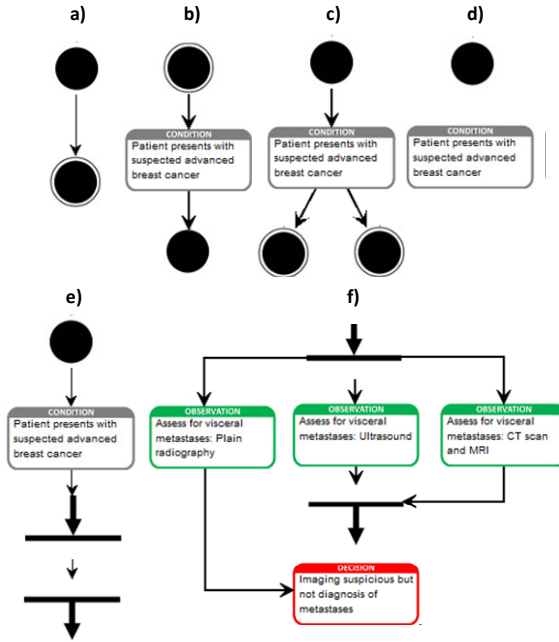


Fig. 2. Some ICs for CIGs encoding the NICE guideline for Advanced Breast Cancer

5 Conclusions

To date, the attempts involved to detect malformed, incomplete, or even inconsistent implementations of computerized guidelines have been not widely applied in practice, since none of these solutions is concerned with directly embedding the theoretic semantics of a formal language as the basis of a guideline formalism in order to easily and directly support its verification.

As a consequence of that, this paper proposed a formal framework, seamlessly embedded into the guideline model, named GLM-CDS, previously conceived by the authors and described in [7]. This framework hybridizes the theoretic semantics of ontology and rule languages to formalize clinical knowledge in the form of a process-like model and, contextually, specify a set of integrity constraints built on the top of it in order to help to detect violations, errors and/or missing information. In particular, it has been thought to preserve the semantics of a hybrid knowledge base, which implements GLM-CDS as an expressive model by combining DLs and LP under OWA, and, in addition, to support a local CWA by formulating ICs as special rules to be

checked against the hybrid knowledge base. The strength of such a solution relies on the hybridization of knowledge representation formalisms for the computerization of a guideline model, which could support guideline developers non-experts in formal methods to construct well-formed CIGs.

In order to promote and facilitate the widespread use of GLM-CDS by health information technology professionals, on-going activities are being carried out to design and realize an ad-hoc, intuitive and user-friendly authoring tool for graphically encoding CPGs and verifying their well-formedness. Finally, the proposed solution will be refined to model integrity constraints pertaining high-level medical properties, concerning, for instance, the exclusion of dangerous or conflicting treatments and the inclusion of the most proper treatments for a considered class of patients.

References

1. Institute of Medicine, *Crossing the Quality Chasm: A New Health System for the 21st Century*. National Academy Press, Washington, DC (2001)
2. Field, M.J., Lohr, K.N.: *Guidelines for Clinical Practice: From Development to Use*. Institute of Medicine, National Academy Press, Washington, DC (1992)
3. Sonnenberg, F.A., Hagerty, C.G.: Computer-interpretable clinical practice guidelines: Where are we and where are we going? In: Kulikowski, C., Haux, R. (eds.) *IMIA Yearbook of Medical Informatics 2006. Methods Inf. Med.*, vol. 45(suppl. 1), pp. 145–158 (2006)
4. Minutolo, A., Esposito, M., De Pietro, G.: KETO: A Knowledge Editing Tool for Encoding Condition–Action Guidelines into Clinical DSSs. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) *HAIS 2012, Part III. LNCS (LNAI)*, vol. 7208, pp. 352–364. Springer, Heidelberg (2012)
5. Pérez, B., Porres, I.: Authoring and verification of clinical guidelines: A model driven approach. *Journal of Biomedical Informatics* 43(4), 520–536 (2010)
6. Bottrighi, A., Giordano, L., Molino, G., Montani, S., Terenziani, P., Torchio, M.: Adopting model checking techniques for clinical guidelines verification. *Artif. Intell. Med.* 48(1), 1–19 (2010)
7. Iannaccone, M., Esposito, M., De Pietro, G.: A Standards-based Verifiable Guideline Model for Decision Support in Clinical Applications. In: *Proc. of the Joint International Workshop KR4HC/ProHealth 2013* (2013)
8. Health Level 7. HL7 Virtual Medical Record (vMR) Project Wiki, [http://wiki.hl7.org/index.php?title=Virtual_Medical_Record_\(vMR\)](http://wiki.hl7.org/index.php?title=Virtual_Medical_Record_(vMR))
9. Hommersom, A., Lucas, P.J., Van Bommel, P.: Checking the quality of clinical guidelines using automated reasoning tools. *Theory Pract. Logic Program.* 8(5-6), 611–641 (2008)
10. Shiffman, R., Greenes, R.: Improving clinical guidelines with logic and decision-table techniques. *Med. Decision Making* 14, 245–254 (1994)
11. Quaglini, S., Saracco, R., Stefanelli, M., Fassino, C.: Supporting tools for guideline development and dissemination. In: *Proc. of Artificial Intelligence in Medicine*, pp. 39–50 (1997)
12. Miller Jr., D.W., Frawley, S.J., Miller, P.: Using semantic constraints to help verify the completeness of a computer-based clinical guideline for childhood immunization. *Comp. Meth. Prog. Biomed.* 58, 267–280 (1999)

13. Duftschmid, G., Miksch, S.: Knowledge-based verification of clinical guidelines by detection of anomalies. *Artif. Intell. Med.* 22, 23–41 (2001)
14. Isern, D., Moreno, A.: Computer-based execution of clinical guidelines: a review. *Int. J. Med. Inform.* 77, 787–808 (2008)
15. Schmitt, J., Hoffmann, A., Balsler, M., Reif, W., Marcos, M.: Interactive Verification of Medical Guidelines. In: Misra, J., Nipkow, T., Sekerinski, E. (eds.) *FM 2006*. LNCS, vol. 4085, pp. 32–47. Springer, Heidelberg (2006)
16. Hommersom, A.J., Groot, P., Lucas, P.J.F., Balsler, M., Schmitt, J.: Verification of medical guidelines using background knowledge in task networks. *IEEE Transactions on Knowledge and Data Engineering* 19(6), 832–846 (2006)
17. Ten Teije, A., Marcos, M., Balsler, M., van Croonenborg, J., Duelli, C., van Harmelen, F., Lucas, P., Miksch, S., Reif, W., Rosenbrand, K., Seyfang, A.: Improving medical protocols by formal methods. *Artificial Intelligence in Medicine* 36(3), 193–209 (2006)
18. Bäumlner, S., Balsler, M., Dunets, A., Reif, W., Schmitt, J.: A verification of medical guidelines by model checking – A case study. In: Valmari, A. (ed.) *SPIN 2006*. LNCS, vol. 3925, pp. 219–233. Springer, Heidelberg (2006)
19. Groot, P., Hommersom, A., Lucas, P., Serban, R., ten Teije, A., van Harmelen, F.: The role of model checking in critiquing based on clinical guidelines. In: Bellazzi, R., Abu-Hanna, A., Hunter, J. (eds.) *AIME 2007*. LNCS (LNAI), vol. 4594, pp. 411–420. Springer, Heidelberg (2007)
20. Prez, B., Porres, I.: Authoring and verification of clinical guidelines: A model driven approach. *Journal of Biomedical Informatics* 43(4), 520–536 (2010)
21. Regenstrief Institute, Inc and the LOINC Committee. Logical Observation Identifiers Names and Codes (LOINC), <http://loinc.org/>
22. International Health Terminology Standards Development Organisation. Systematized Nomenclature of Medicine (SNOMED), <http://www.ihtsdo.org/snomed-ct/>
23. Health Level 7. HL7 Reference Information Model, Version 3, <http://www.hl7.org/implement/standards/rim.cfm>

Hippocampus Localization Guided by Coherent Point Drift Registration Using Assembled Point Set

Anusha Achuthan, Mandava Rajeswari, and Win Mar @ Salmah Jalaluddin

Computer Vision Research Lab, School of Computer Sciences,
Universiti Sains Malaysia, 11800 Penang, Malaysia
{anusha,mandava}@cs.usm.my, salmah@kb.usm.my

Abstract. This paper presents a new approach for hippocampus localization using pairwise non-rigid Coherent Point Drift registration method. The concept of assembled point set is introduced, which is a combination of the available training point sets into a single data space that represents its distribution. Non-rigid Coherent Point Drift is then adapted to register the assembled point set with a randomly chosen base model for hippocampus localization. The primary focus of this work is on the computational intensiveness of the localization approach, in which the proposed localization approach using assembled point set is compared with an existing groupwise non-rigid Coherent Point Drift (GCPD) approach. The computation intensiveness of the proposed approach grows at a quadratic rate as compared with GCPD that grows at a cubic rate. The proposed approach is validated with hippocampus localization task using 40-datasets. The Root Mean Square (RMS) distance between the approximated hippocampus locations and the ground truth is within an acceptable average of 0.6957-mm.

Keywords: point set registration, localization, hippocampus.

1 Introduction

Hippocampus segmentation has received widespread attention from the neuroimaging community [11][9]. Despite the large number of reported success, effective segmentation is still a challenge. This is partly due to the nature of hippocampus, that is relatively smaller in size as compared to adjacent brain structures. Simultaneously, hippocampus also has faint edges and overlapping intensities with adjacent structures, making segmentation even more problematic. As sole reliance on intensity properties is insignificant, supplementing a priori knowledge describing the approximate location of the hippocampus can therefore greatly assist the segmentation task. This allows segmentation to be concentrated directly on the specific region of interest, while avoiding irrelevant brain regions/structures.

Thus far, various localization approaches have been proposed to identify the location of brain structures, which may be categorized into manual, spatial

relation-based, atlas-based and statistical shape model-based approaches. Manual and spatial relation-based localization approaches are highly dependent upon localization information provided by the user, either in the form of mouse clicks [8][18] or a set of predefined rules modelled as fuzzy sets [16][15]. An automatic localization of a target structure was pioneered through atlas-based approaches, which extrapolates information from an atlas to a target dataset using registration procedure [9][17]. However, atlas-based approaches do not capture the information about the range of possible space a structure may occupy on a given population. This information is very important for localizing medical structures with varying geometric properties, such as size and shape between subjects. As an example, the shape and size/volume of the hippocampus tends to differ between healthy control subjects and subjects with neurological disorders [24].

Therefore, efforts have concentrated on adapting Statistical Shape Model (SSM) to localize brain structures due to its credibility in capturing shape variability that may be present within a training population [21]. SSM learns all the available shapes from a training set, and parameterizes the mean/approximate shape and possible shape variations within the population. Besides describing geometric information, the mean shape also provides the approximate location of a shape. Thus, the localization of a target structure is found by exploiting the mean shape, that contains knowledge on the approximate location of the structure [5].

In SSM, shape is most widely represented using the Point Distribution Model (PDM) that describes a shape with a set of landmark points on the structure's boundary [22]. This set of points is perceived to be the salient points on the boundary, and is often referred to as *point set*. The major difficulty in constructing SSM is to find the best correspondence between point sets, especially for three dimensional data. An accurate correspondence assignment is very important to ensure the subsequent process of constructing the mean shape and its variations.

Originally, a well-defined one-to-one correspondence is manually identified between point sets [14][13], followed by Principal Component Analysis (PCA) [10] to construct SSM. This approach is a time consuming and operator subjective procedure, and is thus very impractical to be applied for building SSM of small structures such as the hippocampus. An automatic correspondence assignment between points was then proposed using the Iterative Closest Point (ICP) approach. ICP assigns correspondences to points from two point sets that are closest in Euclidean distance [19][7]. The main drawback of the aforementioned point correspondence assignment methods is the use of only distance measure during correspondence assignment. These methods do not capture any geometric properties between points. As such, good approximation of shape may only be achieved if the training datasets are almost identical in shape and size. Therefore, these methods are not applicable to brain structures, which have been shown to exhibit different shapes and sizes between normal subjects and those with pathologies.

As a solution to this issue, a point set registration method known as Coherent Point Drift (CPD) [2] that includes geometric constraints between neighbouring points during the correspondence assignment has been introduced. The CPD, pioneered by Myronenko and Song [2] is a pairwise point set registration aimed at providing solution for robust alignment of two point sets, and has proven to outperform most state-of-the-art point set registration methods in point sets with outliers or missing points. It assigns automatic correspondences based on probability density estimation theory, that estimates the relative correspondence between two point sets.

The CPD method assumes one of the point sets as an initial *base* model representing Gaussian Mixture Model (GMM) centroids and the other point set as the data points. Correspondences between points are achieved by iteratively fitting the GMM centroids to the data points by minimizing a maximum likelihood function using Expectation-Maximization (EM). An additional regularization term that forces neighbouring points to move coherently is imposed in the CPD function to ensure geometric properties of the shape are preserved. At the optimum, CPD is able to align both point sets to an approximated position, and find the correspondences simultaneously.

Myronenko and Song [2] have implemented the CPD method to solve rigid, affine and non-rigid transformations between two points sets. Non-rigid transformation is more robust for hippocampus registration considering the characteristics of hippocampus that varies in shape and size among subjects. Rigid and affine transformations are more efficient for structures that are almost consistent in shape and size [23]. With proven robustness of the CPD registration method in achieving accurate point set alignment, this proposed work focuses on the adaptability of the pairwise non-rigid CPD method to approximate the shape of the hippocampus.

The goal of this work is to capture available prior localization knowledge from a set of training point sets through the pairwise non-rigid CPD method and generate an approximated shape of the hippocampus. The obtained approximate shape will then be utilized for reflecting the approximate location of a targeted hippocampus. In this proposed approach, the technique of *assembled point set* is introduced, in which the available training point sets are assembled together within a single data space. The main idea of assembling point sets within a single data space is to analyze the distribution patterns of the corresponding training population. The proposed approach of using the pairwise non-rigid CPD with the assembled point set technique will be henceforth referred to as the Assembled-based CPD (ACPD).

Recently, a groupwise non-rigid CPD approach was proposed by Rasoulian et. al. [3] to construct SSM of the hippocampus. Their work involves a preprocessing stage of rigidly registering all the training point sets, k to a common coordinate space of a randomly chosen base model, followed by the pairwise non-rigid CPD registration between the base model with every training point sets, resulting in k number of approximated point sets. Finally, the final mean shape is obtained by recursively updating the chosen base model with the k approximated point

sets using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) Quasi-Newton optimization technique. Despite the similar aim of constructing the mean/approximate shape of the hippocampus, the proposed ACPD is significantly different from [3] in its approach. Therefore, the main focus of this paper is to evaluate and compare the computation intensiveness of both, ACPD and GCPD approaches. Comparison on complexity with [3] shows that the proposed ACPD exhibits lower computational complexity while achieving comparable performance. Experiments validating the robustness of ACPD demonstrates that the proposed ACPD is also able to deliver an acceptable level of localization accuracy.

The remainder of the paper is organized as follows. Section 2 lays down the foundation of the pioneering work of the pairwise non-rigid CPD registration method, followed by Section 3 describing the proposed localization approach. Section 4 and Section 5 evaluates the proposed ACPD approach in terms of complexity and accuracy, respectively. Section 6 concludes the paper with remarks regarding future work.

2 Overview of Pairwise Non-rigid CPD

The pairwise non-rigid CPD registration method [2] formulates alignment and correspondence assignment between two point sets concurrently in a probability density framework. Relating two point sets through GMM density function, one of the point sets is represented as GMM centroids, $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_M)^T$ and the second point set is represented as data points, $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^T$. The CPD method is aimed at achieving optimal correspondence assignment between two point sets by spatially transforming a GMM centroid, \mathbf{y}_M to a new spatial locations through fitting of the centroids to the observed data points, \mathbf{X} . This eventually leads to the alignment of point set, \mathbf{Y} to an optimum positions, \mathbf{T} which represents the final approximated positions of points in the mean shape. The alignment of two point sets is casted as an objective function, $E(\mathbf{Y})$ expressed as:

$$E(\mathbf{Y}) = E_{Data}(\mathbf{Y}) + \frac{\lambda}{2} E_{Reg}(\mathbf{Y}), \quad (1)$$

where $E_{Data}(\mathbf{Y})$ indicates the probability density function of a particular data point, \mathbf{x}_n for M number of centroids. It is defined as a weighted sum of M Gaussian component densities given as:

$$E_{Data}(Y) = p(\mathbf{x}) = \sum_{m=1}^M \frac{1}{M} p(\mathbf{x}|m), \quad (2)$$

where $p(\mathbf{x}|m) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp^{-\frac{\|\mathbf{x}-\mathbf{y}_m\|^2}{2\sigma^2}}$. D is the dimension of the point set and the GMM is assumed to be comprised of equally weighted Gaussians with equal membership probabilities of $P(m) = \frac{1}{M}$ and equal isotropic covariances σ^2 .

The second term in Equation 1, $E_{Reg}(\mathbf{Y})$ is the regularization term that imposes smooth displacement between neighbouring points to maintain the geometric properties of centroids, \mathbf{Y} . The displacement, which specifies the rate a

point may change its position is defined by the displacement function, v . The regularization of the displacement function, $\|Lv\|$ is performed by regularizing the norm of v defined by

$$\|Lv\|^2 = \int_{R^D} \frac{|\tilde{v}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} d\mathbf{s}, \quad (3)$$

where function \tilde{v} indicates Fourier transform of v in the frequency domain, \mathbf{s} using Gaussian kernel, G with standard deviation, β . \tilde{G} is its Fourier transform. The parameter β defines the strength of coherency between points.

Using Equations 2 and 3, the objective function $E(\mathbf{Y})$ is rewritten as:

$$E(\mathbf{Y}) = - \sum_{n=1}^N \log \sum_{m=1}^M P(m) p(\mathbf{x}_n|m) + \frac{\lambda}{2} \|Lv\|^2. \quad (4)$$

where λ controls the amount of regularization.

Restating point set alignment in term of the displacement function, v shows that the centroids, \mathbf{Y} are updated to new positions, \mathbf{T} by adding the initial centroids' position with the displacement function, v , which is described as:

$$\mathbf{T}(\mathbf{Y}, v) = \mathbf{Y} + v(\mathbf{Y}). \quad (5)$$

Hence, in order to solve the point set alignment problem, an optimal value of function v need to be estimated. This estimation is performed by maximizing the likelihood or equivalently by minimizing the negative log-likelihood of the objective function, $E(\mathbf{Y})$ given in Equation 4 using the EM algorithm [1]. The EM algorithm proceeds in two steps as follows:

1. **E-Step:** The expected posterior probability distribution, $P^t(m|\mathbf{x}_n)$ is computed using an initial random estimate of parameters, v^t and σ^t at time t . The posterior probability distribution describes the probability of a centroid \mathbf{y}_m being responsible for generating data point \mathbf{x}_n . Utilizing Bayes' theorem, the posterior probability is formulated as:

$$P^t(m|\mathbf{x}_n) = \frac{\exp^{-\frac{1}{2} \left\| \frac{\mathbf{x}_n - (\mathbf{y}_m + v^t(\mathbf{y}_m))}{\sigma^t} \right\|^2}}{\sum_{k=1}^M \exp^{-\frac{1}{2} \left\| \frac{\mathbf{x}_n - (\mathbf{y}_m + v^t(\mathbf{y}_m))}{\sigma^t} \right\|^2}}. \quad (6)$$

2. **M-Step:** With the known value of $P^t(m|\mathbf{x}_n)$ given in Equation 6, the objective function, $E(\mathbf{Y})$ is redefined in term of the expectation of the complete negative log-likelihood function, Q given as:

$$Q(\mathbf{Y})^{t \rightarrow t+1} = \frac{1}{2\sigma^2} \sum_{n=1}^N \sum_{m=1}^M P^t(m|\mathbf{x}_n) \|\mathbf{x}_n - (\mathbf{y}_m + v(\mathbf{y}_m))\|^2 + \frac{ND}{2} \log \sigma^2 + \frac{\lambda}{2} \|Lv\|^2. \quad (7)$$

With the redefined objective function, $Q(\mathbf{Y})$, the optimal solution for function v is found by minimizing Equation 7 using calculus of variation, which is performed by taking the functional derivative of Q with respect to function v . In the actual implementation, the minimization of Equation 7 to find the optimal value of function v is described to be a linear combination of applying the Gaussian kernel, G on each of the centroids, y_m . The Gaussian kernel is modelled as a kernel matrix, $G(y_i, y_j) = \exp^{-\frac{1}{2} \left\| \frac{y_i - y_j}{\beta} \right\|^2}$. Thus, the optimal value of function v may be obtained through the following function:

$$v = \frac{1}{\sigma^2 \lambda} \sum_{n=1}^N \sum_{m=1}^M P^t(m|\mathbf{x}_n) \|\mathbf{x}_n - (\mathbf{y}_m + v(\mathbf{y}_m))\| G. \quad (8)$$

Then, the new positions of $\mathbf{T}(\mathbf{Y}, v)$ are found by

$$\mathbf{T}(\mathbf{Y}, v) = \mathbf{Y} + \mathbf{G}\mathbf{W}, \quad (9)$$

where $\mathbf{W} = \frac{1}{\sigma^2 \lambda} \sum_{n=1}^N \sum_{m=1}^M P^t(m|\mathbf{x}_n) \|\mathbf{x}_n - (\mathbf{y}_m + v(\mathbf{y}_m))\|$.

Both the E-Step and M-Step are alternated until the EM algorithm reaches convergence. The finally obtained value of \mathbf{T} reflects the final approximated positions. These points are defined to be the points on the boundary of the approximated shape.

3 Proposed Localization Approach Using Assembled Point Set

In the context of this paper, a training point set is perceived to be a group of salient points on the boundary of a manually delineated hippocampus volume. All of the training point sets are assembled together within a single data space to form the assembled point set. For a given k number of training sets, X^k , the assembling process concatenates k points sets into a single point set, \mathbf{S}_{Ns} defined by

$$\mathbf{S}_{Ns} = \parallel_{k=1}^K X_{Nk}^k, \quad (10)$$

where \parallel represents the concatenation process, Ns is the number of points in point set, \mathbf{S} and Nk is the number of points in each training point set.

The assembled point set, \mathbf{S} represents the data points and is utilized in the pairwise non-rigid CPD. The implementation of proposed ACPD involves a random selection of a training point set to be assigned as the base point set, \mathbf{Y} . Then, rigid alignment is performed between all the remaining training point sets with base point set, \mathbf{Y} to align the training points sets to a common coordinate system. Next, the standardized remaining point sets are assembled together to form the point set, \mathbf{S} . Finally, the pairwise non-rigid CPD registration is performed between point set, \mathbf{S} and base point set, \mathbf{Y} . The detailed pseudo algorithm of ACPD is presented in Algorithm 1.

The finally updated point set, \mathbf{T} obtained from the proposed ACPD represents the approximated average positions of the entire training points. These approximated positions are perceived to be the approximate location of salient points on the hippocampus boundary. Hence, the resulting point set, \mathbf{T} is defined to be an approximated prior localization knowledge for an unseen hippocampus volume.

Algorithm 1. The proposed Assembled-based CPD (ACPD)

```

1: Notations:
2:    $\mathbf{X}^k$ : k-th training point set
3:    $N_k$ : Number of points in k-th training point set
4:    $\mathbf{Y}_M$ : Base point set with M centroids
5:    $N_s$ : Number of points in assembled point set, S
6:    $n$ : n-th point
7:    $m$ : m-th centroid

Require:  $K$  number of training point sets,  $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2 \dots \mathbf{X}^K\}$ 

8: function ACPD( $\mathbf{X}$ )
9:   Random selection of a base point set,  $\mathbf{Y}_M$  from training point sets,  $\mathbf{X}$ .
10:  Rigid alignment of the remaining  $K - 1$  training point sets with  $\mathbf{Y}$  to standardize
    all point sets to a common coordinate space.
11:  Concatenation of standardized remaining  $K - 1$  training point sets into single
    point set,  $\mathbf{S}$ .

12:  while convergence do
13:    E-Step: Compute  $P^t(m|\mathbf{S}_n)$  using Equation 6.
14:    M-Step: Using point set,  $\mathbf{S}$  as the data points,
15:      Calculate the displacement value,  $v$  using Equation 8.
16:      Update the new positions of centroids,  $\mathbf{T}$  using Equation 9.
17:  end while
18:  Assign the finally obtained point set,  $\mathbf{T}$  as the approximated shape.
19: end function

```

4 Complexity Analysis

This section compares the complexity of the proposed ACPD with an alternative localization approach using groupwise non-rigid CPD (GCPD) [3]. The main difference between ACPD and GCPD is the utilization structure of the original non-rigid CPD for registering multiple point sets. As non-rigid CPD is the core process in the registration procedure, the complexity analysis is focused on the computation intensiveness of performing the pairwise non-rigid CPD. Figure 1 compares the core components of ACPD and GCPD, in which the pairwise non-rigid CPD has been utilized.

From the highlighted core components of the approaches, the proposed ACPD involves two main processes, which are the assembling of point sets with complexity of $O(N_S)$ and the pairwise non-rigid CPD with complexity of $O(N_S M)$. Thus,

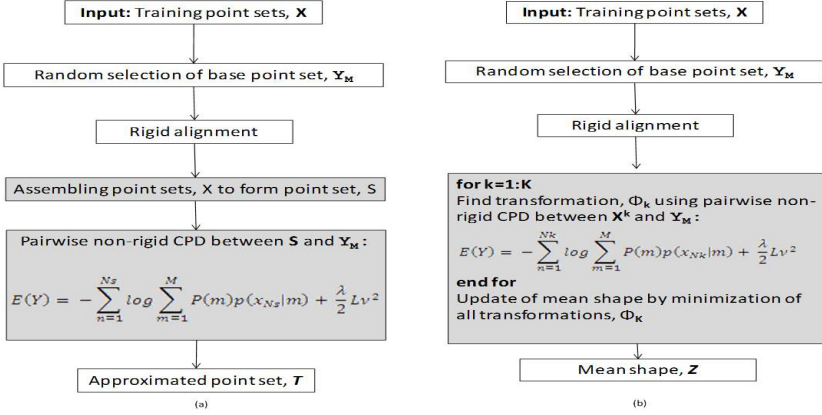


Fig. 1. The core components of ACPD and GCPD

the computation intensiveness of proposed ACPD is $O(N_S + N_S M)$, whereas for GCPD the computation intensiveness is $O(K N_K M)$. This summarizes that the computation intensiveness of the proposed ACPD grows at a quadratic rate as compared with GCPD that grows at a cubic rate. Thus, for a very large number of training point sets, the proposed ACPD is proven to be with less computation intensiveness than GCPD.

5 Experimental Results

A public database that consists of 40 manually delineated hippocampus volumes are used in this study [4]. Each volume contains a set of 2D binary images corresponding to the manually delineated hippocampus region in axial view. A 3D training point set is perceived to be a group of salient points on the boundary of a manually delineated hippocampus volume. These salient points are computed by applying the Canny edge detector [6] on each of the binary images and extracting the resulting local maxima edge points. Then, the 3D training point set is obtained by stacking up all the resulting salient points from all of the 2D image slices. The training point set construction process is illustrated in Figure 2.

The proposed ACPD localization approach is evaluated using the leave-one-out strategy, in which the to-be-localized test subject's hippocampus is left out during the localization process. Root Mean Square (RMS) distances between corresponding points from the ACPD approach and the test subject are used to measure the accuracy of the proposed approach. Figure 3 shows the average RMS values for the 40 datasets using the range of λ values defined in [3]. The value $\beta = 2$ is fixed based on the reported lowest RMS value projected by the experiments conducted on hippocampus in [3]. From this figure, the lowest average



Fig. 2. Construction of a 3D training point set

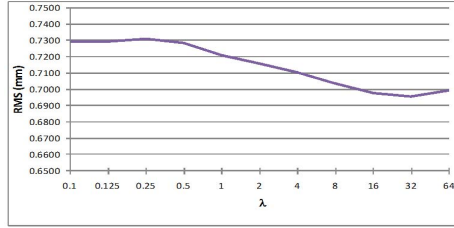


Fig. 3. Average RMS distance of hippocampus with various values of λ and fixed value of $\beta = 2$

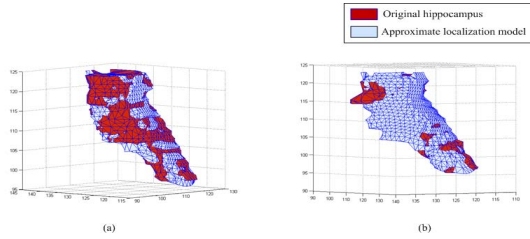


Fig. 4. (a) Hippocampus with lowest RMS distance value of 0.5081 mm. (b) Hippocampus with highest RMS distance value of 0.9621 mm. [$\lambda = 32$]

RMS value of 0.6957 mm is shown when $\lambda = 32$. It may also be concluded that lower values of λ produce higher RMS distances. Lower values of λ indicates that the localization model only adapted a smaller amount of smoothness constraint. Thus, the localization model tends to closely overfit all the training points, instead of points within a specified topological neighbourhood as imposed by the smoothness constraint. Therefore, without sufficient smoothness constraint, the approximated localization model is not able to generalize on unseen hippocampus accurately.

Figure 4 provides the visual comparison of the localized hippocampus exhibiting the lowest and highest RMS distances of 0.5081 mm and 0.9621 mm, respectively. The triangulation of the points sets are performed using the Crust algorithm [12]. From the obtained highest RMS distance, it may be summarized that the proposed ACPD is still able to approximately localize the hippocampus within acceptable values of registration accuracy below 3.5 mm [20].

6 Conclusion

This paper has presented a hippocampus localization approach by utilizing the strength of pairwise non-rigid CPD registration method and adapting it to capture the localization knowledge from a training population using the assembled point set. The proposed ACPD approach has shown to reduce complexity as compared to GCPD [3]. The proposed ACPD approach also demonstrated an acceptable range of RMS distance values below 3.5 mm. Future work will concentrate on testing the ACPD approach on clinical datasets to evaluate the overall localization accuracy.

Acknowledgement. This research is supported by the Ministry of Higher Education (MOHE) Malaysia and Universiti Sains Malaysia (USM) through the RLKA/SLAI scholarship.

References

1. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood estimation from incomplete data via the em algorithm. *Journal of the Royal Statistical Society* 39, 1–38 (1977)
2. Myronenko, A., Song, X.: Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(12), 2262–2275 (2010)
3. Rasouliyan, A., Rohling, R., Abolmaesumi, P.: Group-wise registration of point sets for statistical shape models. *IEEE Transactions on Medical Imaging* 31(11), 2025–2034 (2012)
4. Shattuck, D.W., Mirza, M., Adisetiyo, V., Hojatkashani, C., Salamon, G., Narr, K.L., Poldrack, R.A., Bilder, R.M., Toga, A.W.: Construction of a 3d probabilistic atlas of human cortical structures. *NeuroImage* 39(3), 1064–1080 (2008)
5. Bailleul, J., Ruan, S., Constans, J.M.: Statistical shape model-based segmentation of brain mri images. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2007, pp. 5255–5258 (August 2007)
6. Canny, J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8(6)*, 679–698 (1986)
7. Joseph, J., Warton, C., Jacobson, S.W., Jacobson, J.L., Moltano, C.D., Eicher, A., Marais, P., Phillips, O.R., Narr, K.L., Meintjes, E.M.: Three-dimensional surface deformation-based shape analysis of hippocampus and caudate nucleus in children with fetal alcohol spectrum disorders. *Human Brain Mapping* (2012)
8. Lu, X., Luo, S.: The application of watersnakes algorithm in segmentation of the hippocampus from brain MR image. In: Gao, X., Müller, H., Loomes, M.J., Comley, R., Luo, S. (eds.) MIMI 2007. LNCS, vol. 4987, pp. 277–286. Springer, Heidelberg (2008)
9. Cabezas, M., Oliver, A., Lladó, X., Freixenet, J., Cuadra, M.B.: A review of atlas-based segmentation for magnetic resonance brain images. *Computer Methods and Programs in Biomedicine* 104(3), 158–177 (2011)
10. Sonka, M., Hlavac, V., Boyle, R.: *Image Processing, Analysis, and Machine Vision*. PWS Publishing (1999)

11. Balafar, M.A., Ramli, A.R., Saripan, M.I., Mashohor, S.: Review of brain mri image segmentation methods. *Artificial Intelligence Review* 33, 261–274 (2010)
12. Amenta, N., Bern, M., Kamvysselis, M.: A new voronoi-based surface reconstruction algorithm. In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1998*, pp. 415–421. ACM, New York (1998)
13. Duta, N., Sonka, M.: Segmentation and interpretation of mr brain images using an improved knowledge-based active shape model. In: *Duncan, J.S., Gindi, G. (eds.) IPMI 1997. LNCS, vol. 1230*, pp. 375–380. Springer, Heidelberg (1997)
14. Duta, N., Sonka, M.: Segmentation and interpretation of mr brain images: An improved active shape model. *IEEE Transactions on Medical Imaging* 17(6), 1049–1062 (1998)
15. Colliot, O., Camara, O., Bloch, I.: Integration of fuzzy spatial relations in deformable models: Application to brain mri segmentation. *Pattern Recognition* 39(8), 1401–1414 (2006)
16. Nempont, O., Atif, J., Angelini, E.D., Bloch, I.: Combining radiometric and spatial structural information in a new metric for minimal surface segmentation. In: *Karssemeijer, N., Lelieveldt, B. (eds.) IPMI 2007. LNCS, vol. 4584*, pp. 283–295. Springer, Heidelberg (2007)
17. Coupe, P., Manjon, J.V., Fonov, V., Pruessner, J., Robles, M., Collins, D.L.: Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation. *NeuroImage* 54(2), 940–954 (2011)
18. Yushkevich, P.A., Piven, J., Hazlett, H.C., Smith, R.G., Ho, S., Gee, J.C., Gerig, G.: User-guided 3d active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *NeuroImage* 31(3), 1116–1128 (2006)
19. Besl, P.J., McKay, H.D.: A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(2), 239–256 (1992)
20. Khallaghi, S., et al.: Registration of a statistical shape model of the lumbar spine to 3D ultrasound images. In: *Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010, Part II. LNCS, vol. 6362*, pp. 68–75. Springer, Heidelberg (2010)
21. Heimann, T., Meinzer, H.: Statistical shape models for 3d medical image segmentation: A review. *Medical Image Analysis* 13(4), 543–563 (2009)
22. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Training models of shape from sets of examples. In: *Proc. British Machine Vision Conference 1992*. Springer (1992)
23. Crum, W.R., Hartkens, T., Hill, D.L.G.: Non-rigid image registration: theory and practice. *The British Journal of Radiology* 77, 140–153 (2004)
24. Tae, W.S., Kim, S.S., Lee, K.U., Nam, E.C., Choi, J.W., Park, J.I.: Hippocampal shape deformation in female patients with unremitting major depressive disorder. *American Journal of Neuroradiology* 32(4), 671–676 (2011)

Network Anomaly Classification by Support Vector Classifiers Ensemble and Non-linear Projection Techniques

Eduardo de la Hoz^{1,3}, Andrés Ortiz², Julio Ortega¹, and Emiro de la Hoz^{1,3}

¹ Computer Architecture and Technology Department. CITIC

University of Granada. 18060 Granada, Spain

² Department of Communications Engineering

University of Málaga. 29071 Málaga, Spain

³ Systems Engineering Program

Universidad de la Costa. Barranquilla, Colombia

Abstract. Network anomaly detection is currently a challenge due to the number of different attacks and the number of potential attackers. Intrusion detection systems aim to detect misuses or network anomalies in order to block ports or connections, whereas firewalls act according to a predefined set of rules. However, detecting the specific anomaly provides valuable information about the attacker that may be used to further protect the system, or to react accordingly. This way, detecting network intrusions is a current challenge due to growth of the Internet and the number of potential intruders. In this paper we present an intrusion detection technique using an ensemble of support vector classifiers and dimensionality reduction techniques to generate a set of discriminant features. The results obtained using the NSL-KDD dataset outperforms previously obtained classification rates.

1 Introduction

Network Intrusion Detection Systems (IDS) aim to detect network anomalies by means of analyzing the deviations from the normal behaviour [1]. This way, it is possible to detect unknown attacks without any prior knowledge of new attacks. Usually, IDS calculate some features from the network traffic to be able to classify the traffic, detect abnormal behaviours and react according to some predefined rules. There are two design approaches to IDS [2]. The first consists on looking for patterns corresponding to known signatures of intrusions. The second one searches for abnormal patterns by using more complex features which allow discovering not only an intrusion but also a potential intrusion. Most attacks can be classified into four categories: Denial of Service Attack (*DoS*), Probing Attack (*PROBE*), User to Root Attack (*U2R*) and Remote to Local Attack (*R2L*).

Identifying the specific attack type may be useful in order to react in a specific way instead of only closing ports or connections as preventive response. Nevertheless, detecting not only an attack but also the type is not a straightforward

task, and existing datasets such as KDD99 [3] have inherent problems that require specific techniques for classification. This is specially important for some infrequent attacks, as these databases provide only a reduced number of samples for training. Moreover, not many works in the literature process the available datasets in a proper way, presenting common pitfalls mainly regarding preprocessing, and performance evaluation of the proposed methods [4]. In this work, we proposed a classification technique avoiding these drawbacks, and paying special attention to data preprocessing (i.e. normalization) and feature selection stages. In addition, the proposed method uses a support vector classifiers ensemble to leverage the classification performance even for the less common attacks. Hence, specialized classifiers are used to build a multi-expert classification system, and each classifier is trained with a different feature selection in order to boost the detection capabilities for a specific class. This way, Section 2 and 3, respectively, describe the data preprocessing stage and the use of linear and non-linear dimensionality reduction techniques to deal with feature selection. Then, Section 4 describes the classifier architecture and Section 5 provides the details of the experimental setup, and analyses the performance of the procedure through the corresponding experimental results. Finally, Section 5 summarizes the conclusions of the paper.

2 Methods

In this paper, an anomaly detection method using a support vector classifiers and non linear projection techniques is proposed. In order to provide an overall view of the proposal, Figure shows the block diagram of the classification system. As shown in this Figure, the training phase performs data normalization and feature selection to generate a discriminative sets for training the classifiers. Thus, specialized classifiers are trained for each anomaly type, and further combination of these classifiers during the classification phase yields the eventual classification outcomes. Following sections provide details regarding different stages in Figure 1.

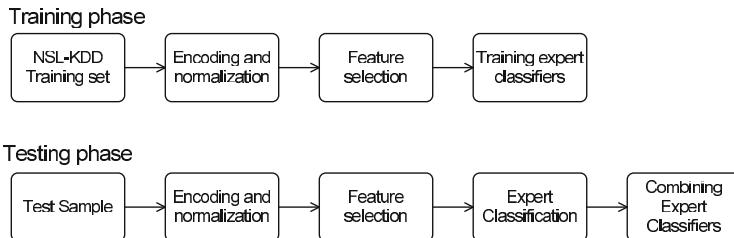


Fig. 1. Block diagram of the classification method

2.1 Data Preprocessing and Normalization

Data preprocessing is an important stage that may determine the classification performance. It comprises encoding non-continuous variables and normalization. Although data normalization plays an important role in the preprocessing stage, as it can determine the classification performance, not many works pay enough attention to it [4]. Moreover, symbolic and binary features have to be treated in a different way. Although symbolic features can be encoded as integer numbers [4,5], it is not the best encoding solution for classifiers based on the euclidean distance [6]. This way, we adopt a different solution that maps each symbolic feature to an \mathbb{R}^d subspace, where d is the number of possible values of the discrete variable.

Data normalization ensures that all the features are in the same scale, in such a way that none of the features contributes more than other in the distance measure. There are different ways to normalize data [7]. In this work, continuous variables are normalized to zero mean and unity variance.

2.2 Feature Selection

After data preprocessing stage, a feature selection process is accomplished with the aim of reducing the dimensionality of the data samples keeping the most discriminative information. In addition, using fewer features reduces the computational burden. Feature selection stage is performed in two steps. In the first step, features with the higher discriminative power for each connection type are preselected using Fisher Discriminant Ratio (FDR), defined as:

$$FDR = \frac{(\mu_i - \mu_j)^2}{\sigma_i + \sigma_j} \quad (1)$$

for the 2-class separation case. In a second step, different dimensionality reduction techniques have been used to show their effectiveness for embedding KDD99-based data. These include linear techniques such as Principal Component Analysis (PCA), and non-linear techniques such as *Kernel PCA* [8] and Isometric Mapping *Isomap* [9].

2.3 Dimensionality Reduction Using PCA

PCA has been widely used in many applications for extracting the most relevant information from a dataset. In fact, it has been successfully used in face recognition applications [10]. In this case, PCA is used to derive a new set of uncorrelated features from a set of correlated ones. Thus, PCA generates a set of orthogonal basis vectors so that the data can be expressed as a linear combination of that basis. Thus, the training data samples are projected onto the subspaces generated by the principal components corresponding to each class, to generate a set of features which best describe each connection. These features are further used to train Support Vector Machines (SVMs). In order to classify a new data instance v_i , it has to be projected onto each subspace, obtaining the corresponding feature vectors.

2.4 Dimensionality Reduction Using Kernel PCA

The main drawback of PCA consists in assuming a linear relationship between features. Thus, it is a strictly linear technique and it is not able to detect non-linear dependences. *Kernel PCA* [8] makes use of the so-called *kernel trick* [7] to solve this drawback, projecting the original data into a high-dimensional space through a (usually non-linear) kernel function. Basically, it may be implemented by substituting all dot products by the kernel function in PCA.

2.5 Dimensionality Reduction Using Isomap

Isomap [9] is a widely used non-linear dimensionality reduction technique that aims to embed the original data into a low-dimensional space preserving pairwise distance between points in the projection space. This is achieved by replacing the Euclidean distance metric in the original space by the geodesic distance, approximated by the shortest path through local proximity matrix. Isomap algorithm can be summarized in three main steps: 1) compute neighbours for each data point, 2) compute pairwise distance matrix (M) between point and 3) find eigenvectors of M.

3 Classification Using Support Vector Classifiers Ensemble

Support Vector Classifiers (SVC) were introduced by Vapnik [11] as a set of useful tools for classification and regression. SVCs seek for the optimal separation hyperplane in a higher dimensional space while maximizing the margin between the hyperplane and the training vectors (i.e. maximum-margin hyperplane is selected). Moreover, it is possible to create non-linear classifiers by means of the *kernel trick*, which replaces every dot product by a non-linear kernel function. Thus, given a dataset $\{x_1, \dots, x_n\}$ and its corresponding class labels $\{y_1, \dots, y_n\}$, SVCs aim to solve the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (2)$$

subject to constraint

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad (3)$$

where ϕ defines the kernel function transforming training vectors x_i into a higher dimensional space. In our case, we used a RBF kernel, defined as $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, $\gamma > 0$. Although SVCs were initially designed for binary classification (i.e. $y \in \{-1, +1\}$), it is possible to extend them for multiclass classification using the *one-against-one* approach [12,13]. However, all the binary classifiers are trained using the same feature set in this case. In this work we used expert classifiers, each specialized in detect a specific attack type. Subsequently, these binary classifiers are combined to produce the overall classification outcome.

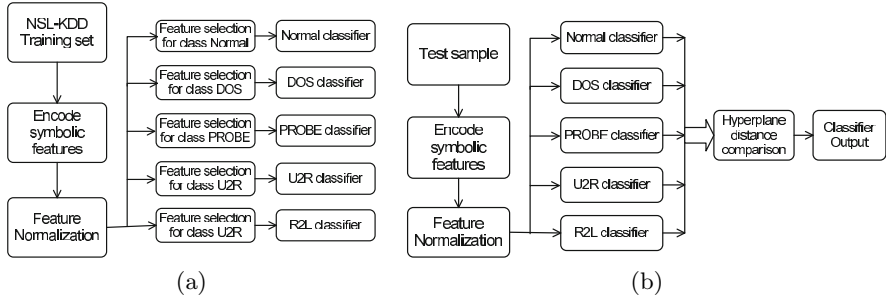


Fig. 2. Training (a) and testing (b) phase

Thus, specific features computed from each attack type in the training set are used to train a binary classifier specialized in differentiate a specific attack from the rest, as shown in Figure 2a. Testing phase is accomplished as follows. When a new sample arrives, it is divided into 5 different flows corresponding to the previously trained expert classifiers. Specific features corresponding to each attack type are extracted in each flow, using the FDR computed features (preselection) and then, these features are projected into the corresponding subspace as described in Section 2.2. Subsequently, each classifier provides a binary output (i.e. indicating whether the sample belongs to the class being detected by the expert classifier or not). In the case of two or more experts classified the same sample as belonging to different classes, the class corresponding to the maximum distance to the SVC hyperplane is selected. This process is shown in Figure 2b.

4 Experimental Setup and Results

The proposed method to detect network anomalies has been evaluated by using cross-validation. Training and testing data is provided by the NSL-KDD [3,14,15] as separate datasets. Thus, it is not necessary to extract subsets for cross-validation assessment from the database.

4.1 Database

The KDD'99-based datasets [3] has been widely used in research works as it contains about 4GB of compressed data from captures of *tcpdump* [14] in the DARPA'98 IDS evaluation program [15]. It corresponds to about 7 weeks of network traffic, and three groups of features, extracted for each connection: Basic features, Traffic-based features and Content-based features, described by 41 features each. Depending on the specific features, continuous, symbolic or binary values are used to encode them. However, the KDD'99 dataset has inherent problems due to the synthetic characteristic of the data [15,3], partially solved

in the Knowledge Discovery and Data Mining (NSL-KDD) dataset which is used in this work as in [16,17].

4.2 Experimental Results and Discussion

In this section, experimental results using the proposed classification scheme are provided. Moreover, we performed a comparison among different linear and non-linear feature reduction techniques in order to identify the technique which provides higher classification accuracy values. Figure 3a shows the classification accuracy obtained for normal connections and 3b, 3c, 3d, 3e for DOS, PROBE, U2R and R2L attack types, respectively as a function of the number of features used to describe each class (i.e. this corresponds to the dimensionality of the projection subspace computed by the techniques indicated in figure legends).

In the case of KernelPCA, polynomial kernel with $d=3$ has been used in the experiments. On the other hand, Radial Basis Function kernel is used in SVC with $\sigma = 5$. As shown in Figure 3, non-linear techniques perform slightly better than linear PCA. However, although linear and non-linear techniques provide similar results for detecting normal connections, non-linear projection techniques present better separating abilities for DOS, PROBE, U2R and R2L attack types using fewer components. In addition, *isomap* provides better results for the same number of components. This can be seen for instance, in Figure 3c, 3d and 3e, where 5 components are enough to provide accuracy values up to 0.9, whereas KernelPCA or linear PCA requires 8-10 components to reach this accuracy value. ROC curve for the two-class case (normal/attack) using Kernel PCA as dimensionality reduction method is shown in Figure 4.

Table 1. True positive and true negative rates for different classification methods

Method	Number of features	True Positive Rate	False Positive Rate
DM-Naïve Bayes [16]	41	96.5	3.0%
Proposed method	23	93.4%	14%
Random Forest [18]	41	80.67	**
Decision Trees [18]	41	81.05	**

** Data not provided by the author

In order to show the *true positives* and *true negatives* ratios that allows identifying false positive ratios, Table 1 shows sensitivity and specificity values for each technique and attack type. Nevertheless, attacks of different types should be evaluated separately for correct performance evaluation [4]. Hence, we computed the ROC curves derived from each expert classifier, obtaining Areas Under respective ROC curves of 0.94, 0.86, 0.93, 0.81 and 0.91 for Normal connections, DOS, PROBE, U2R and R2L respectively.

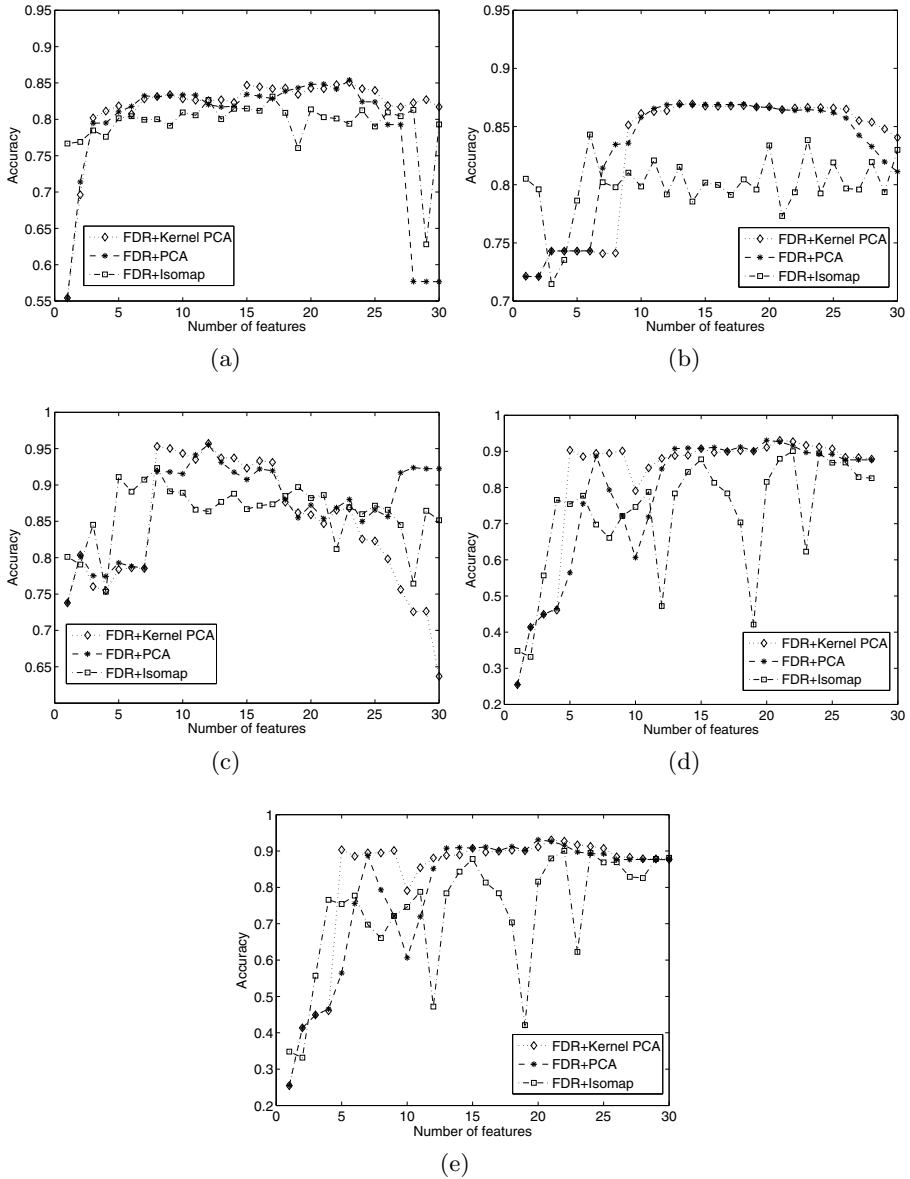


Fig. 3. Classification accuracy for different number of selected features. (a) for normal connections. (b), (c), (d) and (e) for DOS, PROBE, U2R and R2L attacks, respectively.

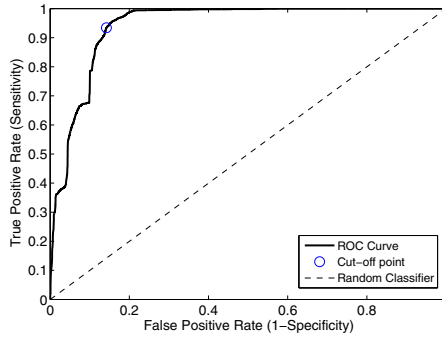


Fig. 4. ROC curve for normal/attack classification using KernelPCA as dimensionality reduction method

5 Conclusions and Future Work

In this paper we present a classification approach for network anomaly classification that combines non-linear dimensionality reduction techniques and an SVC ensemble to build expert classifiers. This provides an efficient method for feature selection obtaining more discriminative features. Moreover, feature preselection stage avoids the use of the less discriminative features or features containing very similar values for all the data instances. Subsequently, three different dimensionality reduction techniques have been used to assess their ability to generate discriminative features for each attack type. These include linear PCA and non-linear techniques such as Kernel PCA and Isomap. The new feature spaces derived using these techniques are used to train the SVC ensemble. In addition, SVCs are not combined with a majority-voting scheme but experts SVCs trained to detect specific attacks are coalesced to determine the class of new test samples. Experiments conducted using the NSL-KDD dataset show that embeddings using non-linear techniques allows discriminating among attack types using fewer components than linear ones, specially for attacks. In addition, *isomap* provides better results for the same number of components. Specifically, 5 *isomap* components are enough to provide accuracy values up to 0.9, whereas KernelPCA or linear PCA requires 8-10 components to reach this accuracy value. Moreover, U2R accuracy results indicate certain database dependence due to the lower number of samples corresponding to this attack.

As future work we plan to improve the method optimizing parameters in both dimensionality reduction and SVC classifiers. Additionally, as our system was designed to classify samples into 5 classes, the two-classes performance is not as high as in other approaches as [16], but this can be solved including more classifiers in the ensemble and optimized features for the two-class case.

Acknowledgments. This work has been funded by the Ministerio de Ciencia e Innovación of the Spanish Government under Project No. TIN2012-32039 and by the University of Málaga. Campus de Excelencia Andalucía Tech.

References

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. *ACM Computing Surveys* 41(3) (2009)
2. Hoffman, A., Schimitz, C., Sick, B.: Intrusion detection in computer networks with neural and fuzzy classifiers. In: Kaynak, O., Alpaydm, E., Oja, E., Xu, L. (eds.) *ICANN 2003 and ICONIP 2003*. LNCS, vol. 2714, pp. 316–324. Springer, Heidelberg (2003)
3. Network Security Lab - Knowledge Discovery and Data Mining (NSL-KDD) (2007), <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
4. Tavallaee, M., Stakhanova, N., Ghorbani, A.: Toward credible evaluation of anomaly-based intrusion-detection methods. *Trans. Sys. Man Cyber Part C* 40, 516–524 (2010)
5. Kayacik, H., Zincir-Heywood, A., Heywood, M.: A hierarchical som-based intrusion detection system. *Journal Engineering Applications of Artificial Intelligence* 20(4), 439–451 (2007)
6. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In: *Applications of Data Mining in Computer Security*. Kluwer (2002)
7. Theodoridis, S., Koutroumbas, K.: *Pattern Recognition*. Academic Press (2009)
8. Müller, K., Mika, S., Ratsch, G., Tsuda, B., Schölkopf, B.: An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks* 12(2), 181–201 (2003)
9. Tenenbaum, J., de Silva, V., Langford, J.: A global geometric framework for non-linear dimensionality reduction. *Science* 290, 2319–2323 (2000)
10. Turk, M., Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3(1), 71–86 (1992)
11. Vapnik, V.N.: *Statistical Learning Theory*. Wiley-Interscience (1998)
12. Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Jackel, L., LeCun, Y., Muller, U., Sackinger, E., Simard, P., Vapnik, V.: Comparison of classifier methods: A case study in handwriting digit recognition. In: *Proc. International Conference on Pattern Recognition*, pp. 77–87 (1994)
13. Bredensteiner, E., Bennett, K.: Multicategory classification by support vector machines. *Computational Optimization and Applications* 12(1-3), 53–79 (1999)
14. Lippmann, R., Fried, D., Graf, I., Haines, J., Kendall, K., McClung, D., Weber, D., Webster, S., Wyszogrod, D., Cuningham, R., Zissman, M.: Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation. *Descex* 2, 1012–1027 (2000)
15. McHugh, J.: Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection systems evaluation as performed by lyncoln laboratory. *ACM Transactions on Information and Systems Security* 3(4), 262–294 (2000)
16. Panda, M., Abraham, A., Abraham, M.: Discriminative multinomial naïve bayes for network intrusion detection. In: *6th Conference on Information Assurance and Security*, IAS (2010)
17. Nziga, J.: Minimal dataset for network intrusion detection systems via dimensionality reduction. In: *6th International Conference on Digital Information Management*, ICDIM (2011)
18. Tavallaee, M., Bagheri, E., Wei, L., Ghorbani, A.: A detailed analysis of the kddcup 1999 dataset. In: *Proceedings of the IEEE International Symposium on Computational Intelligence in Security and Defense Applications CISDA* (2009)

Classification Method for Differential Diagnosis Based on the Course of Episode of Care

Adrian Popiel¹, Tomasz Kajdanowicz¹, Przemyslaw Kazienko¹, Jean Karl Soler², Derek Corrigan³, Vasa Curcin⁴, Roxana Danger Mercaderes⁴, and Brendan Delaney⁵

¹ Institute of Informatics, Wrocław University of Technology, Wrocław, Poland
tomasz.kajdanowicz@pwr.wroc.pl

² Mediterranean Institute of Primary Care, Attard, Malta

³ Department of General Practice, Royal College of Surgeons in Ireland, Ireland

⁴ Department of Computing, Imperial College London, United Kingdom

⁵ Kings College London, United Kingdom

Abstract. The main goal of the paper is to propose a classification method for differential diagnosis in primary care domain. Commonly, the final diagnosis for the episode of care is related with the initial reason for encounter (RfE). However, many distinct diagnoses can follow from a single RfE and they need to be distinguished. The new method exploits the data about whole episodes of care quantified by individual patients' encounters and it extracts episode features from electronic health record to learn the classifier. The experimental studies carried out on two primary care dataset from Malta and the Netherlands for three distinct diagnostic groups revealed the validity of the proposed approach.

Keywords: Classification, Differential Diagnosis Classification, Episode of Care Diagnosis.

1 Introduction

Modern-day human is surrounded by a lot of information. However in short time he learnt how to: use it, put in data warehouses, explore, mine and make profit on it. As far as we deal with these techniques on everyday basis in management and marketing, they are rarely used in primary care.

It is addressed in the paper an interesting problem of differential diagnosis classification that can support the decisions of General Practitioners (GPs). This problem concerns the diagnosis for diseases, which have common reason for the first encounter but end with different, in some cases very serious diagnosis.

This paper provides a proposal of classification method for differential diagnosis in three groups of diseases, for which decision support has a great value for a General Practitioner. The method takes into account medical history encapsulated within the episode of care and results with final episode diagnosis. In order to infer the diagnosis the method uses direct attributes that describes patient (age group, referrals to primary care and specialists) as well as attributes derived from patient episode of care history (for instance number of distinct diagnosis and average number of reasons for encounters of various types).

Reason for encounter(RfE) is a factor that starts encounter with GP. It could be in a form of symptom or complaint, also request for an intervention for example prescription, advice or referral, in some cases even diagnosis from previous encounter can be a RfE. However, it needs to be underlined that in interpretation RfE differs from diagnosis. In particular RfE starts an encounter and diagnosis is a final result of a medical treatment in this encounter[1].

The paper provides concise presentation of related work in the field of medical decision support in Section 2 and description of decision support problem for General Practitioners in Section 3. It is proposed a classification method for differential diagnosis accompanied with short presentation of used algorithms for Differential Diagnosis as well as methods of data evaluation in Section 4. Then experimental results and comparison of the methods are gathered in Section 5 and concluded in Section 6.

2 Related Work

The general source of information for medical decision making is an electronic health record (EHR). In the recent years in the United States it was invested nearly 50 billion dollars to create EHRs as such data repositories are expected to improve quality care and reduce costs. One of the main goals of EHRs is to improve the quality of diagnosis and overall treatment providing EHR based information technology tools, especially decision support systems. Differential diagnosis in primary care might be complex and in some cases requires long diagnostic process. Providing decision support systems might help General Practitioners feel more comfortable and supported in quick diagnostics. Yet practitioners might appreciate additional knowledge source especially when they must make critical diagnosis. Nevertheless, decision support systems have today several number of limitations, but implemented with practical and evidence based approach, can be an important enhancement in primary care decision making process [2].

However, there exist some opposite conclusions like these stated by Romano and Stafford who provided conclusions that EHR is not associated with better quality in health centres, so that make concerns about the ability of health information technology to provide better quality in primary care [3].

Partial answer for that problem brings Korteisto, who described process of implementation of decision support and medical data recording system in health center in Finland [4]. It was highlighted that such systems must be adopted well by employees of health centres in order to bring additional value. Even though the decision support system was introduced to GPs and made familiar for them, practitioners recorded reasons for encounters (RfEs) or diagnosis when patient has already left. This caused wrong qualification of RfEs and errors in data filling. Moreover, in the proposed system it was in fact too much support from the system and it was hard to make profit from such amount of knowledge.

Important part of medical decision support systems is technology, methods and algorithms behind them. Yoo et al. bring great analysis of data mining techniques used

in health care [5]. In their work there were presented data mining techniques and algorithms and as well as contexts of their usage in biomedical and healthcare studies.

Li et al. focused on efficient discovery of risk patterns in medical data[6]. Their algorithm quickly and efficiently discovers cohorts of patients that are vulnerable to a risk outcome. They also compared their method with decision trees and association rules that showed that discovered by their method risk patterns had much more quality than patterns brought by classic data mining methods.

Yang and Wang used Random Forest to design new classifier that create more reliable classes, because in medicine risk of misjudgment is very costly. Their classifier take into account cost of misjudgment and predefined confidence level for each class [7].

Summarizing, the problem of differential diagnosis was rarely touched on in the context of information technology solutions. Especially nobody bind this topic with course of episode of care.

3 Problem Description

In many electronic health record systems (EHRs) gathering information for primary health care, the data records are composed of three main parts: (1) demographic data about the patient (age, sex, region, population), (2) provenance of data (lineage of the processes in a EHR system) and (3) series of following patient encounters. Every such a sequence starts at the first visit when the patient provides his or her primary (initial) reason for encounter (RfE) and terminates with the last encounter containing the final diagnosis, Fig. 1.

From the reasoning point of view the most important episode features are: (1) initial reason for encounter (RfE) expressed by the patient at the first encounter, (2) diagnoses, symptoms and procedures registered during all intermediate visits and (3) the final diagnoses fixed at the last encounter – it can already be fixed during the previous encounters. Both RfEs and diagnoses in primary health care can be coded by means of ICPC2 - International Classification of Primary Care, 2nd Edition developed by WONCA International Classification Committee (WICC) [8], available at <http://icpc.who-fic.nl/browser.aspx>. The main difference between RfE and a diagnosis is their creator: reasons for encounter are provided by patients themselves, whereas diagnoses are made the general practitioners (GPs). The initial RfE remains static for the whole episode but the diagnosis may change even at each encounter according to new physician findings.

Based on the ICPC2 coding schema some aggregated episode of care features can be derived from the component encounters, Fig. 2, see Sec. 4.2 for details.

The main goal of differential diagnosis is to distinguish final diagnoses that can be initiated by a given reason for encounter. In other words, the main goal of analysis of data sets collected in electronic health records for primary care is discover general rules that would help family doctors to be aware of various results (final diagnoses) at the time when a patient tells their complains (RfE), Fig. 2. Hence, differential diagnoses distinguish various final diagnoses that can be caused by single reason for encounter.

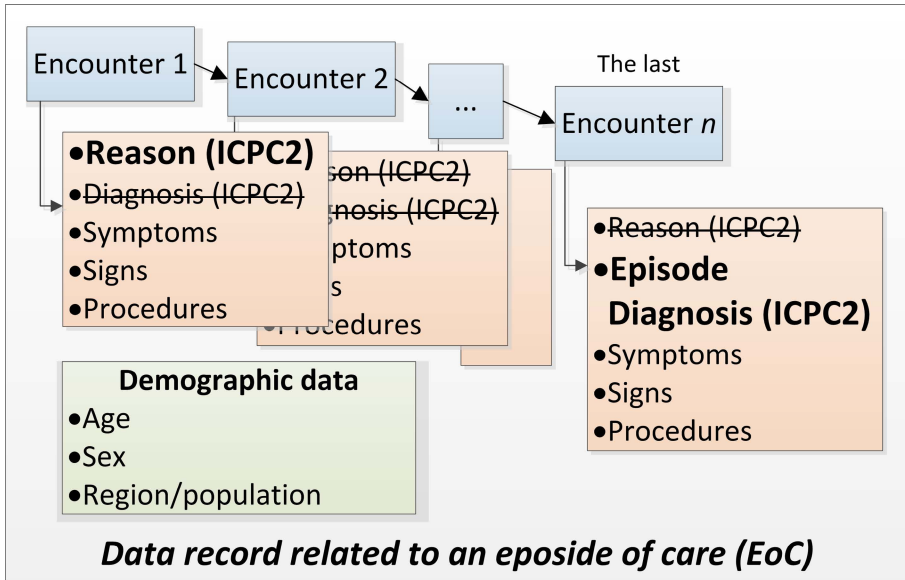


Fig. 1. The data-driven organisation of an episode in primary health care

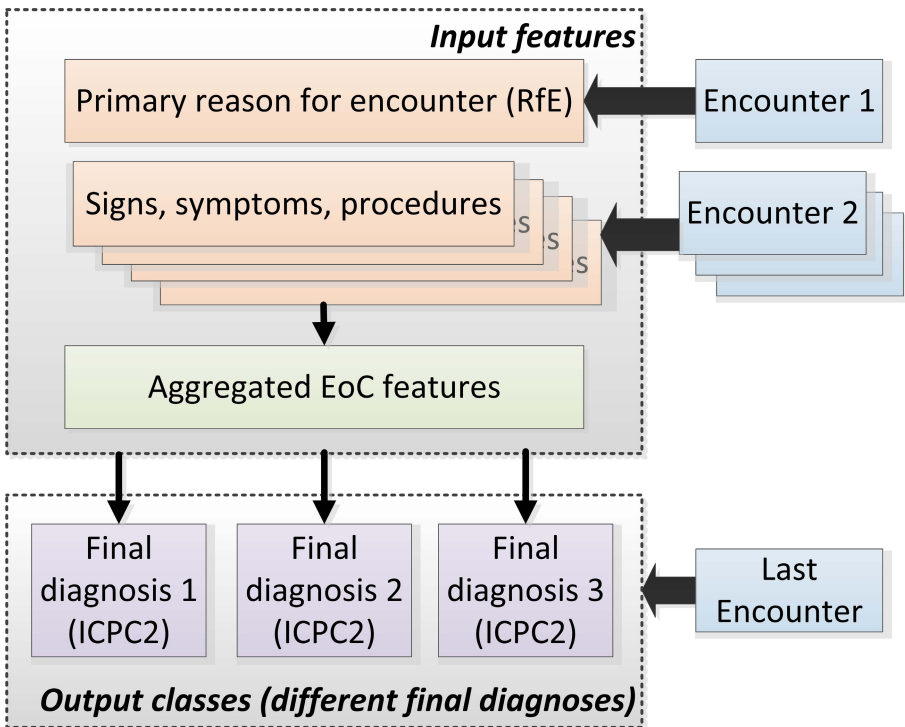


Fig. 2. Differential diagnosis for a single reason for encounter (RfE)

4 Classification Method for Differential Diagnosis

The proposed method for differential diagnosis classification is based on the data extracted from the electronic health record database (EHR). It contains records on individual patient encounters that can be aggregated into single episode of care records (EoC), see Fig. 1 and 2.

The entire classification method for differential diagnosis consists of five major steps, Fig. 3. These are: (1) identification of whole episodes of care (EoC) from the series of individual encounters taken from source EHR (in experimental studies TransHIS data was used as the source data set); (2) extraction of features describing EoC, including primary/initial reason for encounter (RfE) that starts episode of care (in order to select subjects that are taken into account), final diagnosis, i.e. diagnosis from the last encounter used for the classification output; (3) feature selection, (4) learning of the classifier (building the model) and (5) validation (at research) or testing (in decision support system). The method utilizes the original medical data of encounters arranged in episodes of care. In general, the encounter data contains diagnosis (Dia) that was assigned at the encounter by GP alongside with reasons for encounter (RfE) told by the patient, symptoms (Anam) and procedures (Proc) that were prescribed before a given encounter. The information about the encounter is followed by demographics of patient (age group, sex, etc.) as well as referrals to specialists to be undertaken after the encounter. Multiple encounters constitute an episode of care. The episode of care is quantified with final diagnosis called episode diagnosis (EpisodeDia). Moreover, depending on the type of episodes they can have distinct status: new or pre-existing, where an old problem is presented to GP. The method considers only new episodes of care. Moreover, the data describing EoC should have common quantification for RfE, symptoms, diagnosis and procedures, e.g. ICPC2 (International Classification of Primary Care 2nd Edition [8]). The following steps are described below more in-depth.

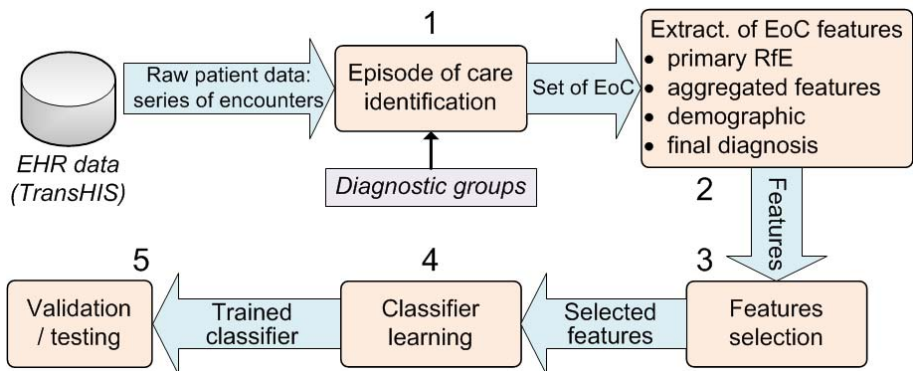


Fig. 3. The genral schema of classification method for differential diagnosis

4.1 Initial RfE, Diagnostic Groups

Based on raw data (consisting of encounters arranged in episodes of care) only new episodes of care were selected (step 1 in Fig. 3). Fortunately, TransHIS data set provides an indicator of episode status. In other cases this would require additional episode status discovery. Since the main purpose of the method is to distinguish diagnosis for some selected primary reason for encounter, the episodes were partitioned into three groups, based on their initial RfE (according to ICPC2 codes). These are RfE known as relatively hard to make a correct diagnosis at the beginning (it means that they especially require differential diagnosis):

Group 1, initial RfE: D01 - Abdominal Pain, with final episode diagnosis:

- D73 - Gastroenteritis presumed infection
- D88 - Appendicitis
- D93 - Irritable Bowel Syndrome
- D99 - Disease digestive system, Chrons disease
- U70 - Pyelonephritis/pyelitis
- U71 - Cystitis/urinary infection other
- W80 - Ectopic Pregnancy
- X74 - Pelvic Inflammatory Disease
- X77 - Malignant neoplasm genital other
- X81 - Genital neoplasm other/unspecified, Ovarian Cancer

Group 2, initial RfE: A11 - Chest Pain, with final episode diagnosis:

- A70 - Tuberculosis
- K74 - Ischaemic heart disease
- K93 - Pulmonary Embolism
- R81 - Pneumonia
- R84 - Malignant neoplasm bronchus/lung
- R85 - Malignant neoplasm respiratory, Mesothelioma
- R88 - injury respiratory other, Pneumothorax
- R96 - Asthma

Group 3, initial RfE: R02 - Shortness of breath/dyspnoea, with final episode diagnosis:

- K77 - Heart Failure, Right Ventricular Failure
- K82 - Pulmonary heart disease, Cor Pulmonale
- K83 - Heart valve disease, Aortic Stenosis
- R78 - Acute Bronchitis/bronchiolitis
- R95 - Chronic obstructive pulmonary disease

Experimental studies were restricted only to above three groups.

4.2 Episode of Care Aggregated Features

Commonly, the basic form of raw data provides only basic information about episodes such as episode diagnosis, certainty of this diagnosis and initial status. In order to derive more episode specific features, enumeration of encounters within the episode was proposed. Then, some aggregated features describing the whole episode were computed (step 2 on Fig. 3). The following were used for that purpose: the number of specific encounters, their percentage, average, variance, standard deviation, maximum, minimum and median of RfE, symptom and diagnosis within a particular ICPC2 diagnosis group. For instance, the above mentioned aggregations of RfE were calculated for all RfE in the episode with type D (Digestive, ICPC code starting with 'D'), another set of features for type A, B, H, K, etc. In general, this feature derivation will result in 17 (the number of distinct diagnosis groups in ICPC2) attributes for RfE, symptom and diagnosis for each of aggregation method. It provides all together 408 attributes describing each episode.

Moreover, a similar feature derivation can be applied to medical procedures, additionally enumerated in five groups based on procedure number:

1. 30-49 - Diagnostic and preventive procedures
2. 50-59 - Treatment procedures, medication
3. 60-61 - Test results
4. 62 - Administrative
5. 63-69 - Referral and other reasons for encounter

This will result in 595 new attributes. Next, it can be calculated 5 additional attributes about the course of episode of care, in particular the number of RfE's, symptoms, diagnosis and procedures in the episode. In order to quantify the frequency of encounter during the episode there are extracted number, average, variance, standard deviation, maximum, minimum and median of days, weeks and years between encounters in the episode – all together 21 attributes.

Eventually, due to the fact that number of attributes is large, feature selection method is applied (step 3 in Fig. 3) [9]. It happens often that there are rejected maximum, minimum and median values as they are not presenting good discrimination and such a selection was performed within experiments.

5 Experimental Study

5.1 Experimental Scenarios

The main purpose of performed experiments was to evaluate the predictive accuracy of the method for differential diagnosis. Moreover it was expected to observe whether results of the method can be used by general practitioners. For the experiments, the tool KNIME[10] version 2.7.3 with Weka[11] 3.6 was used. As classifiers the following models were utilized: J48 (C4.5), Random Forest (RF) and Naive Bayes (NB). In order to learn and validate these classifiers (they were learnt with default WEKA parameters), 10-fold cross-validation procedure was applied (steps 4 and 5 in Fig. 3). For evaluation purpose, the F1-score (also named as F-measure or F-score) was used. It includes both

precision and recall to generate the score for the model. To define precision and recall in our case, there is a need to provide such values as: true positives (tp), false positives (fp) and false negatives (fn). Terms positives and negatives are related to classifier results, whereas true and false values are connected with prediction verified by some kind of external observation. So we can describe true positive as correct result in both ways, true negative as expected misjudgement (or lack of result), false positive represents correct but unexpected prediction and false negative as straightforward missing result.

Hence, precision is defined as follows [9]:

$$precision = \frac{tp}{tp + fp}, \quad (1)$$

recall as [9]:

$$recall = \frac{tp}{tp + fn}, \quad (2)$$

and F1-score as:

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad (3)$$

5.2 Datasets

The experiments were performed on TransHIS data set collected in the Netherlands[1]. There were extracted three groups of episodes of care as described in Section 4.2. Group 1 consisted of 28 thousand of episodes, Group 2 and Group 3 contained 39 thousand of episodes. The data sets were processed with 10-fold cross validation.

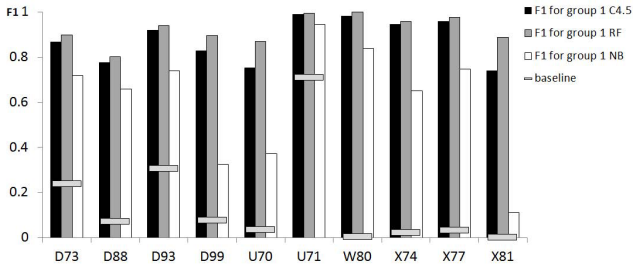
5.3 Results

The experimental result are presented in Figures 4a, 4b and 4c. Vertical axis represents F1-score. Horizontal axis presents diagnosis from proper group. The gray mark on all figures represents the baseline level of F1-score. This is a random baseline above which the differential diagnosis outperform random classification.

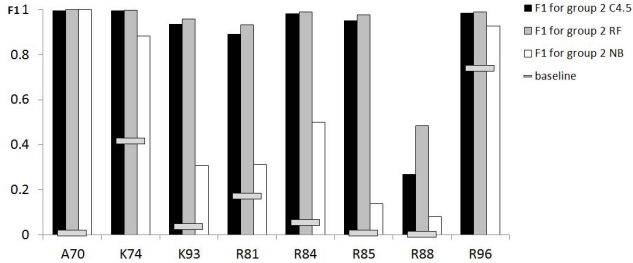
In almost all cases Naive Bayes classifier (NB) returns much less satisfying results, but that was expected since it is the simplest algorithm with two very optimistic assumptions. C4.5 and Random Forest provide quite similar results, even though Random Forest is slightly better. However, C4.5 has a great advantage: it provides human understandable rules that can be interpreted by physicians. That is why C4.5 results are treated as the main achievement of the research. It can be directly used in primary health care as help for general practitioners in their diagnosis.

For group 1, Fig(a) all algorithms distinguished all diagnosis. All of the classifiers resulted with higher F1-score than a baseline. The same situation exists in group 2(b) and group 3(c). The worst results were obtained with Naive Bayes classifier.

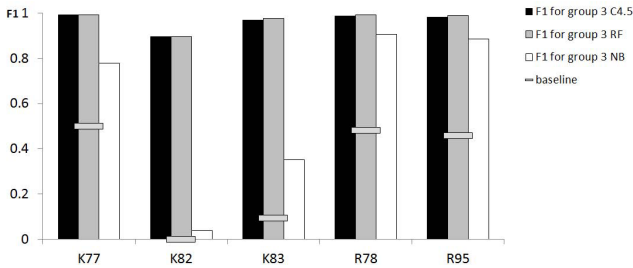
The best results were achieved for group 3. However, for diagnosis K82, Naive Bayesian classifier provided results close to the random baseline. The proposed method always provides a good prediction, for K82 itself, but there were too few episode diagnosis with K82 (only 13 cases, so there were only 13 true positives). The algorithm



(a) Group 1



(b) Group 2



(c) Group 3

Fig. 4. Results of differential diagnosis for all three algorithms (decision tree C4.5, random forest - RF, naive Bayes - NB) for distinct diagnostic groups

also made much more predictions as K82 for other diagnosis - K77 and K88 (so, there were much more false positives: as of 620). As a result, precision (eq. 1) was low and F1-score (eq. 3) remained close to the baseline.

6 Conclusions and Future Work

A new method for differential diagnosis in primary health care was proposed in the paper. Appropriate modelling of source data (especially aggregated feature extraction) facilitated achieving good results in distinguishing diagnosis for a given reason for

patient encounter. Commonly, the best quality results were obtained for the random forest classifier even though C4.5 decision tree was not much worse.

The results can support general practitioners in their diagnosis making based only on simple and general reason for encounter like 'abdominal pain' or 'chest pain'.

Future work will focus on analysis of other diagnostic groups, their medical verification as well as application of ensemble classification methods verified on other data sets.

Acknowledgement. This work was partially supported by The Polish National Center of Science the research project 2011-2014 and the European project TRANSFoRM, FP7 247787, 2011-15, www.transformproject.eu.

References

- [1] Soler, J.K., Okkes, I.: Reasons for encounter and symptom diagnoses: a superior description of patients' problems in contrast to medically unexplained symptoms (mus). *Family Practice* 29(3), 272–282 (2012)
- [2] Schiff, G., Bates, D.: Can electronic clinical documentation help prevent diagnostic errors? *The New England Journal of Medicine* 362(12), 1066–1069 (2010)
- [3] Romano, M., Stafford, R.: Electronic health records and clinical decision support systems. *Arch. Intern. Med.* 171(10), 897–903 (2011)
- [4] Kortteisto, T., Komulainen, J., Kunnamo, I., Makela, M., Kaila, M.: Implementing clinical decision support for primary care professionals – the process. *Finnish Journal of eHealth and eWelfare* 4(3), 153–164 (2012)
- [5] Yoo, I., Alafaireet, P., Marinov, M.: Data mining in healthcare and biomedicine: A survey of the literature. *J. Med. Syst.* 36, 2431–2448 (2012)
- [6] Li, J., Fu, A., Fahey, P.: Efficient discovery of risk patterns in medical data. *Artificial Intelligence in Medicine* 45, 77–89 (2009)
- [7] Yang, F., Wang, H.: Using random forest for reliable classification and cost-sensitive learning for medical diagnosis. In: *The Seventh Asia Pacific Bioinformatics Conference* (2009)
- [8] WONCA: An introduction to the international classification of primary care version 2. Technical report, World Organization of Family Doctors (WONCA), WONCA International Classification Committee (WICC) (2004)
- [9] Olson, D., Delen, D.: *Advanced Data Mining Techniques*. Springer (2008)
- [10] Tiwari, A., Sekhar, A.K.T.: Review article: Workflow based framework for life science informatics. *Comput. Biol. Chem.* 31(5-6), 305–319 (2007)
- [11] Frank, E., Hall, M.A., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I.H., Trigg, L.: Weka - a machine learning workbench for data mining. In: Maimon, O., Rokach, L. (eds.) *The Data Mining and Knowledge Discovery Handbook*, pp. 1305–1314. Springer (2005)

Movie Recommendation Framework Using Associative Classification and a Domain Ontology

María N. Moreno*, Saddys Segrera, Vivian F. López,
María Dolores Muñoz, and Angel Luis Sánchez

Department of Computing and Automatic. University of Salamanca
Plaza de los Caídos s/n, 37008 Salamanca
mmg@usal.es

Abstract. The increasing acceptance of web recommender systems is mainly due to improvements achieved through intensive research carried out over several years. Numerous methods have been proposed to provide users with more and more reliable recommendations, from the traditional collaborative filtering approaches to sophisticated web mining techniques. In this work, we propose a complete framework to deal with some important drawbacks still present in current recommender systems. Although the framework is addressed to movies' recommendation, it can be easily extended to other domains. It manages different predictive models for making recommendations depending on specific situations. These models are induced by data mining algorithms using as input data both product and user attributes structured according to a particular domain ontology.

Keywords: Recommender Systems, Semantic Web Mining, Associative Classification, First-rater, Cold-start, Sparsity.

1 Introduction

Web recommender systems are used in many application domains to predict consumer preferences and assist web users in the search for products or services. The methods used to do that have different levels of complexity, ranging from those that recommend products based on associations between them in previous transactions, to those that make recommendations based on evaluations that users provide about products and similarity between user preferences. The latter, known as collaborative filtering (CF) methods, are the most successful; however, some important drawbacks in them have been reported, especially in traditional approaches based on nearest neighbor algorithms, which show serious performance and scalability problems. In addition, the great number of evaluations needed by these methods in order to provide precise recommendations causes the sparsity problem when evaluations from users are insufficient. Improvements deriving from the research carried out over several years are being incorporated to current web systems, yielding more and more

* Corresponding author.

effective recommendations. Data mining algorithms have been applied to deal with sparsity and performance problems since they are not only based on product evaluations but on other attributes. Moreover, they are induced off-line, before the user logs onto the system, and therefore the time spent on building the model has no effect on the user response time. Nevertheless, sparsity can also reduce the precision of data mining by different degrees depending on the type of algorithm. Therefore, it is necessary to find data mining algorithms slightly sensitive to sparsity in order to obtain precise recommendations. In spite of the advantages of data mining methods, there are situations in recommender systems in which it is very difficult to give recommendations to the user. For instance, when new products without evaluations are introduced into the catalog or when a new user without evaluations about products requests recommendations, the first-rater and the cold-start problems arise respectively. In this paper, a recommendation framework is proposed which aims at overcoming the main drawbacks of current recommender systems.

The rest of the paper is organized as follows: Section 2 is devoted to describing the state of the art and the main problems of recommender systems. In section 3 the proposed framework is presented. The framework is validated through a case study reported in section 4, where the ontology for this particular application and a comparative study of the results from different algorithms are also gathered. Finally, the conclusions are given in section 5.

2 Related Work

2.1 Recommender Systems' Methods

Recommendation methods can be classified into two main categories [Lee et al., 2001]: Collaborative filtering (CF) and content-based approach. Techniques in the first category, initially based on nearest neighbor algorithms, are used to predict product preferences for a user based on the opinions of other users.

Currently there are two approaches for collaborative filtering, memory-based (user-based) and model-based (item-based) algorithms. Memory-based algorithms, also known as nearest-neighbor methods, were the earliest used [Resnick et al., 1994]. They treat all user items with statistical techniques in order to find users with similar preferences (neighbors). The prediction of preferences (recommendations) for the active user is based on the neighborhood features. The advantage of these algorithms is the quick incorporation of the most recent information, but the disadvantage is that the search for neighbors in large databases is slow. In order to avoid this inconvenience, model-based CF algorithms have been proposed. They use data mining techniques in order to develop a model of user ratings, which is then employed to predict user preferences. There are a great variety of data mining algorithms that can be applied in model-based CF. Neural networks were the first of this kind of method [Bilgus and Pazzani, 1998], which changed the nearest neighbor approach of CF methods for a classification approach. Bayesian networks constitute another technique widely used in the induction of recommendation models in a single way [Breese et al., 1998], or jointly with other methods [Campos et al., 2010]. The main shortcoming of these

methods is the high computational cost of building the net, especially when the amount of data is great. Support Vector Machines (SVM) can also be used in recommender systems [Xu and Araki, 2006]. In some works, SVM is used as a complementary technique for other methods [Diez et al., 2008].

The works referenced in this section are just a small sample of the numerous data mining proposals to be used in collaborative filtering based recommender systems. However, the current trend, especially in sparse contexts where ratings are insufficient, is to exploit hybrid methodologies combining content-based and collaborative filtering approaches in order to take advantage of the strengths of each of them [Barragáns-Martínez et al., 2010]. In recent works, semantic information is added to the available data in order to formalize and classify product and user features. These works are commented in section 2.3.

Collaborative filtering (CF), especially the memory-based approach, has certain limitations that have an important impact on the quality of the recommendations. First, rating schemes can only be applied to homogeneous domain information. Furthermore, sparsity and scalability are serious weaknesses which would lead to poor recommendations [Cho et al., 2002]. Sparsity occurs when the number of ratings needed for prediction is greater than the number of the ratings obtained because CF usually requires user-explicit expression of personal preferences for products. The second limitation is related to performance problems in the search for neighbors in memory-based algorithms. These problems are caused by the need to process large amounts of information. The computer time grows linearly with both the number of customers and the number of products in the site. The lesser time required for making recommendations is an important advantage of model-based methods. This is due to the fact that the model is built off-line before the active user goes into the system, but it is applied on-line to recommend products to the active user. Therefore, the time spent in building the model has no effects on the user response time since little processing is required when recommendations are requested by the users, contrary to the memory-based methods that compute correlation coefficients when the user is on-line. Nevertheless, model-based methods present the drawback that recent information is not added immediately to the model but a new induction is needed in order to update the model.

Although the drawbacks described above may be minimized by means of data mining methods, there are other shortcomings that may occur even with these methods. The first-rater (or early-rater) problem arises when it is not possible to offer recommendations about an item that was just incorporated into the system and, therefore, has few evaluations (or even none) from users. Analogously, this drawback also occurs with a new user joining the system: since there is no information about his or her preferences, it would be impossible to determine his or her behavior in order to provide recommendations. Actually, this variant of the first-rater problem is also referred to as the “cold-start problem” [Guo, 1997] in the literature. The grey-sheep problem [Claypool et al., 1999] is another drawback associated with collaborative filtering methods. This problem refers to the users who have opinions that do not consistently agree or disagree with any group of users. As a consequence, such users do not receive recommendations.

The problems addressed here have been treated in some works in the literature. One way of dealing with sparsity and scalability problems consists of reducing the dimensionality of the database used for CF by means of a technique called Singular Value Decomposition (SVD) [Vozalis and Margaritis, 2005]. Barragáns-Martínez et al. [2010] have adapted the proposal of Vozalis and Margaritis for a hybrid system combining content-based and CF approaches in the TV program recommendation domain. The cold-start problem has also been addressed in recent works. Most of them focus on finding new similarity metrics [Bobadilla, 2012] for the memory-based CF approach since traditional measures such as Pearson's correlation and cosine provide poor recommendations when the available number of ratings is scant, a situation that becomes critical in the cases of the cold-start and first-rater problems. Hybrid content-based and CF approaches have also been applied to deal with the first-rater problem. As a representative framework we can cite Fusion of Rough-Set and Average-category-rating (RSA), which integrates multiple contents and collaborative information to predict user preferences based on the fusion of Rough-Set and Average-category-rating [Su et al., 2010].

2.2 Associative Classification

A way to deal with the sparsity problem present in recommender systems is to find a method slightly sensitive to data sparsity. Some studies have demonstrated that associative classification algorithms yield higher precision than other data mining algorithms with sparse datasets [Moreno et al., 2010][Pinho et al., 2012]. These algorithms are association algorithms that produce rules, named class association rules (CARs), containing only the class attribute in the consequent part [Liu et al., 1998]. Therefore, they can be used as machine learning algorithms for classification.

A proposal of this category of methods is the Classification Based on Association (CBA) algorithm [Liu et al., 1998], which consists of two parts, a rule generator based on Apriori for finding association rules and a classifier builder based on the rules discovered. Classification Based on Multiple Class-Association Rules (CMAR) [Li et al., 2001] is another two-step method; however, CMAR uses a variant of FP-growth instead of Apriori. Another group of methods, named integrated methods, build the classifier in a single step. Classification Based on Predictive Association Rules (CPAR) [Yin and Han, 2003] is the most representative algorithm in this group.

Associative classification methods are not widely used in recommender systems in spite of their better behavior in sparse data contexts [Pinho et al., 2012].

2.3 Semantic Web Mining

In recent works semantic web mining is used in order to improve recommendations [Blanco et al., 2008, 2010], [Moreno et al., 2010], [Kim et al., 2011]. This approach is a new research field in which the Semantic Web and Web Mining converge. Semantic Web Mining aims at improving the results of Web Mining by exploiting the semantic structures in the web as well as building the semantic web making use of web mining

techniques [Stumme et al., 2006]. We will focus on the first case since it is the target of the research presented in this paper.

Taxonomic abstractions provided by an ontology allow patterns to be induced at a more abstract level, that is, regularities can be found between categories of products instead of between specific products. These patterns can be used in recommender systems for recommending new products that still have not been rated by the users. In [Kim et al., 2011] a new way of enriching user data in recommender systems is proposed. The idea consists of discovering relevant and irrelevant topics for users and generating new tags for building the user model. Semantic information is also used in [Yuan et al., 2013] in order to improve the recommendations in real estate web sites. The proposal consists on a case-based reasoning method used in combination with an ontological structure.

3 Recommendation Framework

The aim of the proposed framework is to overcome some of the main drawbacks of recommender systems: scalability, sparsity, first-rater and cold-start problems. In some of the works in the literature these problems are tackled separately but no proposal is reported to deal jointly with all of them. Our proposal provides a reference frame for making recommendations depending on the circumstances happening at recommendation time. Recommendations are made by predictive models generated by data mining algorithms. Their generation and their application are carried out by two separate processes represented by the two parts of the framework (Figure 1):

1. *Off-line process*: Corresponds to the induction of the predictive models and it is carried out before users are using the system (off-line). These models will be updated periodically in order to incorporate the information of new users and new products as well as the ratings that users give to the products.
2. *On-line process*: In charge of making recommendations to users by checking the models induced in the off-line process. This part is carried out when users request a recommendation (on-line). The process is different for new and old users because new users have not rated any product and therefore their preferences are unknown.

In the off-line process historical information is used for building two different models needed for recommendations:

1. *Low level model*: Relates specific products and users to preference ratings and are used for making recommendation in ordinary situations. Their induction does not require semantic annotations.
2. *High level model*: Relates types of products and types of users to ratings. Products and users must be previously classified according to a taxonomy given by an ontology designed for the particular application area. This model is applied when first-rater and cold-start problems arise, since recommendations are based on characteristics of both products and users but not on their evaluations. Therefore, neither ratings for new products nor ratings from new users are required for checking the model and providing recommendations.

The on-line recommendation process is valid for both new and old users. New users are required to register in order to obtain semantic information about them, which is needed for classifying them according to the ontology. Since new users have not rated any product they only can receive recommendations by means of the high level model. The patterns enclosed in the model that match the values of the user attributes are selected in order to provide products with the characteristics involved in these patterns given that high level models relate attributes of users and products. These products will be recommended to the user. When a user who has rated products (old user) asks for recommendations, both the low level and the high level models are checked in order to find, respectively, rated and non-rated products to recommend to that person. Building the models off-line provides the additional advantage of avoiding scalability problems since the time spent on the induction of the models does not affect the user response time. Moreover, in order to deal with the sparsity drawback, we recommend using algorithms slightly sensitive to data sparsity, such as associative classification methods, which provide better precision than other methods in sparse data contests. This framework has been validated with data from MovieLens, a movie recommender system.

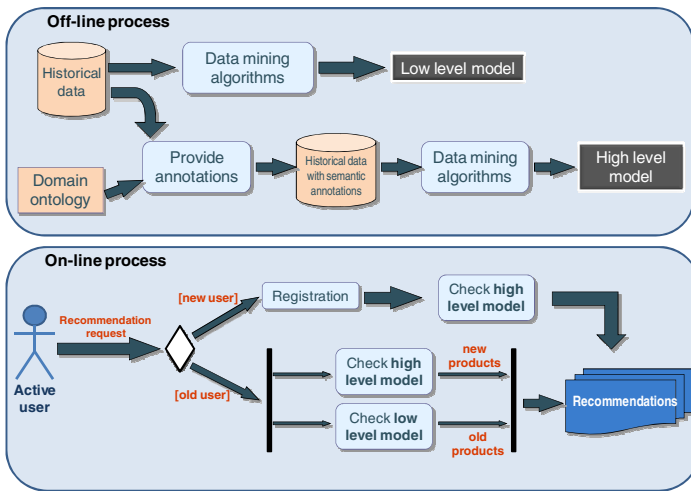


Fig. 1. Recommendation framework

4 Case Study

MovieLens data sets contain user demographic information and user ratings of movies, collected through the MovieLens Web site (<http://movielens.umn.edu>) over a seven-month period. User ratings were recorded on a numeric five-point scale. The rating information consists of 100,000 ratings (1-5) from 943 users on 1682 movies. The comparative study was carried out with a subset of 1,000 records from the database. Since the rating attribute is used to decide whether a movie is going to be recommended to a user, we changed this attribute in order to have only two values:

“Not recommended” (score 1 or 2) and “Recommended” (score 3, 4 or 5). This new attribute, `rating_bin`, will be the label attribute to be predicted. In this way, the classification is simplified and no further transformation is needed for making the recommendations to the user.

All of the users have registered their gender, age, occupation and zip code. The attributes about movies are: title, release date, video release date and another 19 devoted to each possible movie genre or category (unknown, action, adventure, animation, children, comedy, crime, documentary, drama, fantasy, film-noir, horror, musical, mystery, romance, science-fiction, thriller, war and western).

In this work, an ontology is used to improve recommender systems and overcome their main drawbacks, previously commented. Data from MovieLens have been classified and annotated with semantic metadata according to a domain-specific ontology. We have adapted a public ontology about movies from TONES Ontology Repository. Since some of the information of the TONES Ontology is not available in the MovieLens database, the ontology has been simplified. On the other hand, the MovieLens database contains demographic and rating information from users, which is used for making the recommendations. Therefore, we have organized this data according to a different ontology related to user characteristics, such as gender, age and occupation. Finally, the available data taken into account for our application domain was the following:

- User: `id_user`, gender, age, occupation, zip.
- Movie: `id_movie`, title, genre.
- Rating: `id_user`, `id_movie`, score, `rating_bin`.

The definition of the proposed ontology is shown in Figure 2.

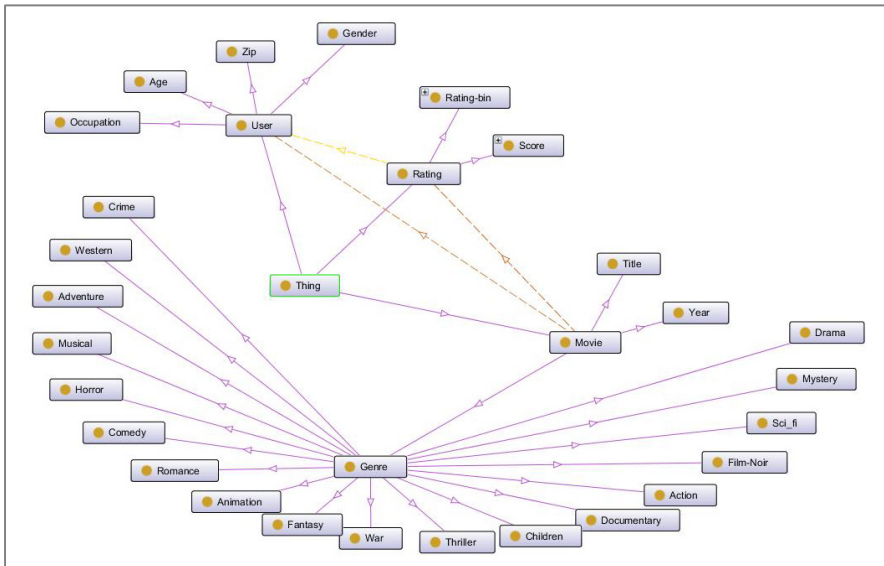


Fig. 2. Specific Ontology for the case study

The target of the proposed framework is to predict user preferences in an efficient way in order to recommend him products that he is interested in. The scenario of recommending rated products to old users has been widely studied and their results are good enough. Therefore, this study is focused on analyzing the precision of high level models where there is a loss of information with regard to the low level models. In this way all tested cases can be considered to be affected by the first-rater and cold-start problem because no rating information is taking into account for making the predictions.

As commented earlier, a way to deal with the sparsity problem is to apply methods that are slightly sensitive to data sparsity; therefore, we try associative classification because of the better behavior of these methods in sparse data contexts [Moreno et al., 2010][Pinho et al., 2012]. Consequently, more reliable recommendations can be obtained with a lesser number of ratings. The studied associative classification algorithms were CBA, CMAR, FOIL and CPAR. They were compared with non-associative classification methods (Decision Tree J48, Bayes Net, Nearest Neighbor and Random Tree) and two multiclassifiers (Bagging and Boosting). The results are shown in Tables 1 and 2.

Table 1. Precision obtained with different data mining algorithms

Algorithms	Precision (%)
Decision Tree J48	72.34
CBA	85.94
CMAR	91.06
FOIL	84.79
CPAR	49.85

Table 2. Precision (%) obtained with three different machine learning algorithms used first as individual classifiers and later as base classifiers in two multi-classifiers

Algorithms	Individual	Bagging	Boosting (AdaBoost)
Bayes Net	81.77	81.61	81.77
Nearest Neighbour	73.54	73.39	73.14
Random Tree	72.34	77.82	75.97

In all experiments 10-fold cross validation was applied. For associative classification methods a support threshold of 20% and a confidence threshold of 80% were used. The associative classification methods CBA, CMAR and FOIL yielded better precision than the non-associative classification algorithms, including multiclassifiers. The best precision was obtained by CMAR (91.06%), one of the associative classification methods. This can be considered a very good result taking into account that models are built at high abstraction level where the available information is less than the low level model information. The CMAR method was also the most

time-consuming; however, this is not a critical disadvantage since the models are induced off-line and the time spent in their building does not influence the user response time. The CBA algorithm can also be a good choice because it shows better precision than the other classifiers and its execution time is similar. It can be used when models require frequent updates. The results confirm the better performance of associative classifiers compared to traditional classifiers in the context of recommender systems where the data is very sparse.

5 Conclusions

Web recommender systems have been the focus of intensive research in the last few years; however, they still involve several shortcomings, such as scalability, sparsity, first-rater and cold-start problems. In this work a recommendation framework especially addressed to overcome these weaknesses is proposed. The proposal consists of combining web mining methods and domain specific ontologies in order to induce models at two abstraction levels. The lowest level models are built from data without semantic information. In contrast, high level models require web data annotated with semantic information according to the defined ontology. This allows patterns to be generated at a high level of abstraction by means of a data mining algorithm. At this level, the models relate types of products and user profiles instead of specific products or particular users. These models are used for recommending non-rated products or for making recommendations to new users, avoiding in this way the first-rater and cold-start problems, respectively.

In addition, the off-line model induction avoids scalability problems in recommendation time and the proposal of using associative classification methods provides a way to deal with the sparsity problem.

References

1. Barragáns-Martínez, A.B., Costa-Montenegro, E., Burguillo, J.C., Rey-López, M., Mikic-Fonte, F.A., Peleteiro, A.: A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Information Sciences* 180, 4290–4311 (2010)
2. Billsus, D., Pazzani, M.J.: Learning collaborative information filters. In: 15th International Conference in Machine Learning, Bari, Italy, pp. 46–54. Morgan Kaufmann (1998)
3. Blanco-Fernández, Y., Pazos-Arias, J.J., Gil-Solla, A., Ramos-Cabrer, M., López-Nores, M., García-Duque, J., Fernández-Vilas, A., Díaz-Redondo, R.P., Bermejo-Muñoz, J.: A flexible semantic inference methodology to reason about user preferences in knowledge-based re-commender systems. *Knowledge-Based Systems* 21, 305–320 (2008)
4. Bobadilla, J., Ortega, F., Hernando, A., Bernal, J.: A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems* 26, 225–238 (2012)
5. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, Madison, Wisconsin, USA, pp. 43–52 (1998)

6. Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Rueda-Morales, M.A.: Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian Networks. *International Journal of Approximate Reasoning* 51, 785–799 (2010)
7. Cho, H.C., Kim, J.K., Kim, S.H.: A Personalized Recommender System Based on Web Usage Mining and Decision Tree Induction. *Expert Systems with Applications* 23(1), 329–342 (2002)
8. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining content-based and collaborative filters in an online newspaper. In: *ACM SIGIR Workshop on Recommender Systems*, Berkeley, CA. ACM Press (1999)
9. Diez, J., del Coz, J.J., Luaces, O., Bahamonde, A.: Clustering people according to their preference criteria. *Expert Systems with Applications* 34, 1274–1284 (2008)
10. Guo, H.: Soap: Live recommendations through social agents. In: *Proc. of Fifth DELOS Workshop on Filtering and Collaborative Filtering*, Budapest, November 10–12 (1997)
11. Kim, H.N., Alkhaldi, A., El Saddik, A., Jo, G.S.: Collaborative user modeling with user-generated tags for social recommender systems. *Expert Systems with Applications* 38, 8488–8496 (2011)
12. Lee, C., Kim, Y.H., Rhee, P.K.: Web Personalization Expert with Combining collaborative Filtering and association Rule Mining Technique. *Expert Systems with Applications* 21, 131–137 (2001)
13. Li, W., Han, J., Pei, J.: CMAR. Accurate and efficient classification based on multiple class-association rules. In: *Proc. of the IEEE International Conference on Data Mining, ICDM 2001*, California, pp. 369–376 (2001)
14. Liu, B., Hsu, W., Ma, Y.: Integration classification and association rule mining. In: *Proc. of 4th Int. Conference on Knowledge Discovery and Data Mining*, pp. 80–86 (1998)
15. García, M.N.M., Lucas, J.P., Batista, V.F.L., Martín, M.J.P.: Multivariate Discretization for Associative Classification in a Sparse Data Application Domain. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) *HAIS 2010, Part I. LNCS (LNAI)*, vol. 6076, pp. 104–111. Springer, Heidelberg (2010)
16. Pinho, J., Segrera, S., Moreno, M.N.: Making use of associative classifiers in order to alleviate typical drawbacks in recommender systems. *Expert Systems with Applications* 39(1), 1273–1283 (2012)
17. Resnick, P., Iacovou, N., Suchack, M., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: *Proc. of ACM CSW 1994 Conference on Computer. Supported Cooperative Work*, pp. 175–186 (1994)
18. Stumme, G., Hotho, A., Berendt, B.: Semantic Web Mining. State of the art and future direction. *Journal of Web Semantics* 4, 124–143 (2006)
19. Su, J.H., Wang, B.W., Hsiao, C.Y., Tseng, V.S.: Personalized rough-set-based recommendation by integrating multiple contents and collaborative information. *Information Sciences* 180, 113–131 (2010)
20. Vozalis, M.G., Margaritis, K.G.: Applying SVD on item-based filtering. In: *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications, ISDA 2005*, pp. 464–469 (2005)
21. Xu, J.A., Araki, K.: A SVM-based personal recommendation system for TV programs. In: *Proc. Int. Conf. on Multi-Media Modeling Conference*, Beijing, China, pp. 401–404 (2006)
22. Yin, X., Han, J.: CPAR. Classification based on predictive association rules. In: *SIAM International Conference on Data Mining, SDM 2003*, pp. 331–335 (2003)

Construction of Sequential Classifier Based on Broken Stick Model

Robert Burduk and Pawel Trajdos

Department of Systems and Computer Networks, Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
`robert.burduk@pwr.wroc.pl`

Abstract. This paper presents the problem of building the sequential model of the classification task. In our approach the structure of the model is built in the learning phase of classification. In this paper a split criterion based on the broken stick model is proposed. The broken stick distribution is created for each column of the confusion matrix. The split criterion is associated with the analysis of the received distributions. The obtained results were verified on ten data sets. Nine data sets come from UCI repository and one is a real-life data set.

Keywords: Broken stick distribution, sequential classifier, confusion matrix.

1 Introduction

Finding the classification rules is becoming more and more difficult when the number of classes in data set increases. For some classifiers the increasing number of classes causes a significant decrease in the quality or loss of performance [1]. One of the possible solutions to this problem is to use multistage classifiers. The general idea of the sequential methods is to break up classification into a number of simple decision [2], [3], [4]. So the built classifiers are usually more flexible than the single-stage classifiers, and their ability to class prediction is generally higher [5], [6], [7].

In particular, this paper discusses a way to design a decision tree structure. The split criterion is based on the confusion matrix. The potential division of the node is associated with the analysis of misclassification in the learning process. In the experiment decision rules are chosen arbitrarily in the entire tree.

The content of the work is as follows. Section 2 introduces the idea of the hierarchical (sequential) classifier. In Section 3 the proposed split criterion is described. In the next section the results of the experiments verified on data sets form UCI repository and one real-life data set of the computer-aided medical diagnosis are presented. The last section concludes the paper.

2 Related Work

Generally, the synthesis of the multistage classifier is a complex problem. It involves a specification of the following components [4], [8]:

- design of a decision tree structure [9],
- selection of features used at each non-terminal node of the decision tree [10], [11],
- the choice of decision rules for performing the classification.

A decision tree structure can be built in two main ways. The first method is to build a tree structure in the learning process [12], [13]. This type of the tree structure induction does not need any additional information about the nature of the problem. However the obtained structure can vary significantly. In other approaches the decision tree structure is fixed before the learning process [14]. However this kind of structure induction needs some expert knowledge.

Due to the number of attributes used for the test carried out in the tree node the univariate and multivariate tests can distinguished. In the univariate methods only one attribute is tested in the tree node. Examples of such algorithms are ID3 [13] and C4.5 [15]. Whereas the multivariate algorithm tests at least two attributes in node. An example of such an algorithm is the CART [16].

The choice of decision rules in nodes can be done in a local or global way. The local choice minimizes node error but does not guarantee minimization of global classifier error. On the other hand the global choice of decision rule guarantees minimum global error but these methods are more computationally demanding than the local ones. The comparison of the above mentioned methods as applied to the Bayesian classifier can be found in [5].

The broken stick model [17] is mostly used in ecology. Experiments conducted in [18] showed that the MacArthur's model is adequate in describing a relative abundance of various species. Moreover the paper [19] affirmed that it can also describe a niche separation. The application of the broken stick model is not limited to ecology. In [20] and [21] it was shown that mentioned model can be utilized in molecular biology. In [20] it was used to estimate the number of possible protein folds. The authors of [21] applied the broken stick model to describe the abundance of amino acids in proteins. The broken stick model can be also adopted in machine learning. Frontier showed in [22] that this model can be used in order to estimate the relevant number of principal components in the PCA method. The values obtained by the broken stick model are compared to the eigenvalues obtained from the PCA. In this case the splitted resource is the total variance of data. However, experimental studies conducted in [23] showed that the broken stick model has a tendency to underestimate the number of relevant principal components.

3 Hierarchical Classifier

The hierarchical classifier contains a sequence of actions [5], [24]. These actions are simple classification tasks executed in the individual nodes of the decision tree. Some specific features are measured on every nonleaf node of the decision tree. At the first nonleaf node features x_0 are measured, at the second features x_1 are considered and so on. Every set of features comes from the whole vector of features. In every node of the decision tree the classification is executed according

to the specific rule. The decisions i_0, i_1, \dots, i_N are the results of recognition in the suitable node of the tree. The design of a decision tree structure is based on the split criterion.

In our task of classification the number of classes is equal to NC . The terminal nodes are labeled with the number of the classes from $M = 1, 2, \dots, NC$, where M is the set of labels classes. The non-terminal nodes are labeled by the numbers of 0, $NC+1$, $NC+2$ reserving 0 for the root-node. The notation for the received model of the multistage recognition can be presented as follows [8]:

- $\overline{\mathcal{M}}$ – the set of internal (nonleaf) nodes,
- \mathcal{M}_i – the set of class labels attainable from the i -th node ($i \in \overline{\mathcal{M}}$),
- \mathcal{M}^i – the set of nodes of the immediate descendant node i ($i \in \overline{\mathcal{M}}$),
- m_i – the node of the direct predecessor of the i -th node ($i \neq 0$).

In each interior node the recognition algorithm is used. It maps observation subspace to the set of the immediate descendant nodes of the i -th node [25], [16]:

$$\Psi_i : X_i \rightarrow \mathcal{M}^i, \quad i \in \overline{\mathcal{M}}. \quad (1)$$

This approach minimizes the misclassification rate for the particular nodes of a tree. The decision rules at each node are mutually independent. In the experiment the decision rules are chosen arbitrarily in the entire tree. Each of the classifiers used in the nodes of the tree takes a decision based on the full set of attributes available in the training set.

In our method of induction, the classification tree is a regular binary tree. This means that on each of the tree nodes there is a leaf or a node which has two children.

Induction of the decision tree is performed by the top-down method. This means that it is initiated by the classifier located in the root of the tree. Using the proposed criterion the decision is made whether to continue the division. The process is repeated for the subsequent child nodes of the tree, until the state wherein the nodes in the tree can no longer be divided.

4 Split Criteria

The broken stick model was proposed in [17]. This model describes the relative abundance of species by random segmentation of a line representing the resources of the environment. Assuming that the unit interval is divided into N spaces of random length C_k , $k \in 1, 2, \dots, N$. Then the expected size of the k -th largest space is:

$$E(C_k) = \frac{1}{N} \sum_{j=0}^{N-k} \frac{1}{N-j}. \quad (2)$$

In the proposed method the division of the internal node is made on the basis of the broken stick distribution in the confusion matrix. Specifically, the broken stick distribution is created for the rows of the confusion matrix. For all class

Table 1. The confusion matrix for the nonleaf node i

		estimated			
		k_1	k_2	\dots	k_L
	k_1	$w_{1,1}$	$w_{1,2}$	\dots	$w_{1,L}$
true	k_2	$w_{2,1}$	$w_{2,2}$	\dots	$w_{2,L}$
	\vdots	\vdots	\vdots	\dots	\vdots
	k_L	$w_{L,1}$	$w_{L,2}$	\dots	$w_{L,L}$

labels from the internal node the $L \times L$ dimensional confusion matrix is created. The example of the confusion matrix is presented in Tab. 1.

The columns of the confusion matrix correspond to the predicted labels (decisions made by the classifier in the internal node). The rows correspond to the true class labels. The $w_{i,j}$ element is the number of i -th class elements classified as the j -th class. In this matrix the diagonal elements represent the overall performance of each label. The off-diagonal elements represent the errors related to each label.

Now the split criterion will be presented. For each class label l the number of misclassified objects is counted:

$$W(k_l) = \sum_{m=1, m \neq l}^L w_{l,m}. \tag{3}$$

Then all the values $W(k_l)$ are normalized:

$$W^*(k_l) = \frac{W(k_l)}{\sum_{i=1}^L W(k_i)} \tag{4}$$

The obtained $W^*(k_l)$ values apportion the total error made by the classifier in node. For these values can therefore use the broken stick model. The values $W^*(k_l)$ are not ascending sort, and compare with the expected values of the broken stick distribution. The values that are greater than the corresponding expected value of the broken stick distribution means that the classifier error is greater than the expected random error. Classes for these labels should be recognized by the next node of sequential classifier.

The division of node occurs when in the values $W^*(k_l)$ we can distinguish both the larger and smaller ones than the corresponding value of the expected broken stick distribution. Otherwise, there is no division of the node. If there is no division at the beginning of the experiment, it indicates that the classification process is performed in the one-stage approach.

5 Experiments

In the experiential research several data sets were tested. The first set refers to the acute abdominal pain diagnosis problem and comes from the Surgical

Clinic Wroclaw Medical Academy. The other nine data sets come from UCI repository [26]. A set of all the available features was used for all data sets, however, for the acute abdominal pain data set the selection of features has been made in accordance with the suggestions from another work on the topic [27], [28]. The numbers of attributes, classes and available examples of the investigated data sets are presented in Tab. 2.

Table 2. Description of data sets selected for the experiments

Data set	example attribute class		
Acute Abdominal Pain	476	31	8
Breast Tissue	106	10	6
Ecoli	336	7	8
Glass Identification	214	10	6
Irys	150	4	3
Lung Cancer	31	52	3
Seeds	210	7	3
Vertebral Column	310	6	3
Wine	178	13	3
Yeast	1484	8	10

Tab. 3 presents the mean error and average ranks for one step classifier. In Tab. 4 we presented the mean error and the average ranks for sequential classifier. The average ranks are calculated on the basis of the Friedman test.

Table 3. Average error for the one-step classifier

Data set	3 - NN	5 - NN	7 - NN	9 - NN	SVM
Acute	0.159	0.161	0.171	0.19	0.182
Breast	0.46	0.475	0.521	0.512	0.342
Ecoli	0.129	0.143	0.137	0.12	0.18
Glass	0.313	0.35	0.362	0.387	0.381
Iris	0.041	0.031	0.029	0.032	0.023
Lung	0.557	0.365	0.593	0.341	0.488
Seeds	0.117	0.102	0.097	0.082	0.094
Vertebral	0.188	0.17	0.165	0.184	0.143
Wine	0.29	0.333	0.301	0.297	0.025
Yeast	0.489	0.432	0.424	0.432	0.448
Aver, rank (group)	3.2	3.1	3.1	3.0	2.6
Aver, rank (all)	6.15	5.39	5.75	5.45	4.94

The value of achieved improvement is not significant from the statistical point of view. For the post-hoc Bonferroni-Dunn test [30], [29] the critical difference (CD) for the 10 algorithms and 10 data sets is equal $CD = 3,76$. This CD is calculated at $\alpha = 0.05$.

Table 4. Average error for the sequential classifier

Data set	$3 - NN^{SBS}$	$5 - NN^{SBS}$	$7 - NN^{SBS}$	$9 - NN^{SBS}$	SVM^{SBS}
Acute	0.157	0.162	0.177	0.184	0.18
Breast Tissue	0.476	0.517	0.556	0.537	0.342
Ecoli R	0.126	0.135	0.133	0.12	0.18
Glass R	0.322	0.36	0.356	0.373	0.376
Iris	0.041	0.031	0.021	0.032	0.023
Lung	0.557	0.554	0.533	0.403	0.488
Seeds	0.117	0.102	0.097	0.082	0.094
Vertebral Column	0.184	0.169	0.168	0.184	0.143
Wine	0.279	0.333	0.3	0.303	0.025
Yeast R	0.481	0.432	0.423	0.427	0.448
Aver, rank (group)	3.2	3.4	2.6	3.1	2.7
Aver, rank (all)	5.76	5.19	4.85	5.85	4.65

Experiments done in the work show that the promising results have been obtained. The proposed approach slightly improved the quality of classification for all except $9 - NN$ classifiers. However the differences are far from the critical difference.

6 Conclusions

In the paper a split criterion based on the analysis of the confusion matrix is proposed. Specifically, the division associated with an incorrect classification is introduced. This criterion is used in the design of a decision tree structure in the multistage classifier. With a fulfilled criteria a binary split of the analyzed decision node is carried out.

The idea of using the resource apportionment models in sequential classification needs to be carefully explored. In order to achieve better results using another statistical model of an error apportionment can be considered. A good starting point is to use other models proposed by MacArthur. On the other hand some improvement can be done by using split criterion that considers the correctly classified objects. An approach that combines the correctly and incorrectly classification rate is worth considering. In another approach to the sequences classification, the separable linearization [31] can be used in the split criterium.

Acknowledgments. The work was supported in part by the statutory funds of the Department of Systems and Computer Networks, Wroclaw University of Technology and by the by The Polish National Science Centre under the grant N N519 650440 which is being realized in years 2011–2014.

References

1. Kolakowska, A., Malina, W.: Fisher Sequential Classifiers. *IEEE Transaction on Systems, Man, and Cyberneics – Part B Cyberneics* 35(5), 988–998 (2005)
2. Mitchell, T.M.: *Machine Learning*. McGraw-Hill Comp., Inc., New York (1997)
3. Podolak, I.T.: Hierarchical classifier with overlapping class groups. *Expert Syst. Appl.* 34(1), 673–682 (2008)
4. Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. *IEEE Trans. Systems, Man Cyber.* 21(3), 660–674 (1991)
5. Burduk, R.: Classification error in Bayes multistage recognition task with fuzzy observations. *Pattern Analysis and Applications* 13(1), 85–91 (2010)
6. Cyganek, B.: Image Segmentation with a Hybrid Ensemble of One-Class Support Vector Machines. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) *HAIS 2010, Part I. LNCS*, vol. 6076, pp. 254–261. Springer, Heidelberg (2010)
7. Wozniak, M., Grana, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. *Information Fusion* (2013), doi: <http://dx.doi.org/10.1016/j.inffus.2013.04.006>
8. Kurzyński, M.: Decision Rules for a Hierarchical Classifier. *Pat. Rec. Let.* 1, 305–310 (1983)
9. Wozniak, M.: A hybrid decision tree training method using data streams. *Knowledge and Information Systems* 29(2), 335–347 (2010)
10. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *The Journal of Machine Learning Research* 3, 1157–1182 (2003)
11. Rejer, I.: Genetic Algorithms in EEG Feature Selection for the Classification of Movements of the Left and Right Hand. In: Burduk, R., Jackowski, K., Kurzynski, M., Wozniak, M., Zolnierok, A. (eds.) *CORES 2013. AISC*, vol. 226, pp. 579–589. Springer, Heidelberg (2013)
12. Penar, W., Woźniak, M.: Experiments on classifiers obtained via decision tree induction methods with different attribute acquisition cost limit. *Advances in Soft Computing* 45, 371–377 (2007)
13. Quinlan, J.R.: Induction on Decision Tree. *Machine Learning* 1, 81–106 (1986)
14. Manwani, N., Sastry, P.S.: Geometric decision tree. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 42(1), 181–192 (2012)
15. Quinlan, R.J.: C4.5: Programs for Machine Learning. *Machine Learning* 16(3), 235–240 (1993)
16. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. John Wiley and Sons (2000)
17. Macarthur, R.: On the relative abundance of bird species. *Proc. Natl. Acad. Sci. USA* 43, 293–295 (1957)
18. King, C.E.: Relative Abundance of Species and MacArthur’s Model. *Ecology* 45(4), 716–727 (1964)
19. De Vita, J.: Niche Separation and the Broken-Stick Model. *The American Naturalist* 114(2), 171–178 (1979)
20. Leonov, H., Mitchell, J.S., Arkin, I.T.: Monte Carlo Estimation of the Number of Possible Protein Folds: Effects of Sampling Bias and Folds Distributions. *Proteins* 51, 352–359 (2003)
21. Yoshiaki, I., Sumie, U., Sumie, H.: The broken-stick model for amino acid composition in proteins. *Journal of Molecular Evolution* 16(1), 69–72 (1980)

22. Frontier, S.: Etude de la décroissance des valeurs propres dans une analyse en composantes principales: comparaison avec le modele du baton brise. *Biol. Ecol.* 25, 67–75 (1976)
23. Cangelosi, R., Goriely, A.: Component retention in principal component analysis with application to cDNA microarray data. *Biol. Direct* 2(2) (2007)
24. Kurzyński, M.: On the Multistage Bayes Classifier. *Pattern Recognition* 21, 355–365 (1988)
25. Berger, J.: *Statistical Decision Theory and Bayesian Analysis*. Springer, New York (1993)
26. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
27. Burduk, R., Woźniak, M.: Different decision tree induction strategies for a medical decision problem. *Central European Journal of Medicine* 7(2), 183–193 (2012)
28. Kurzyński, M.: Diagnosis of acute abdominal pain using three-stage classifier. *Computers in Biology and Medicine* 17(1), 19–27 (1987)
29. Dunn, O.J.: Multiple Comparisons Among Means. *Journal of the American Statistical Association* 56, 52–64 (1961)
30. Derrac, J., Garcıya, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* 1(1), 3–18 (2011)
31. Bobrowski, L., Topczewska, M.: Separable Linearization of Learning Sets by Ranked Layer of Radial Binary Classifiers. In: Burduk, R., Jackowski, K., Kurzynski, M., Wozniak, M., Zolnierek, A. (eds.) *CORES 2013. AISC*, vol. 226, pp. 135–144. Springer, Heidelberg (2013)

Model and Feature Selection in Hidden Conditional Random Fields with Group Regularization

Rodrigo Cilla, Miguel A. Patricio, Antonio Berlanga, and José M. Molina

Computer Science Department. Universidad Carlos III de Madrid
Avda. de la Universidad Carlos III, 22
28270 Colmenarejo (Madrid). Spain
{rcilla,mpatrici}@inf.uc3m.es, {aberlan,molina}@ia.uc3m.es

Abstract. Sequence classification is an important problem in computer vision, speech analysis or computational biology. This paper presents a new training strategy for the Hidden Conditional Random Field sequence classifier incorporating model and feature selection. The standard Lasso regularization employed in the estimation of model parameters is replaced by overlapping group-L1 regularization. Depending on the configuration of the overlapping groups, model selection, feature selection, or both are performed. The sequence classifiers trained in this way have better predictive performance. The application of the proposed method in a human action recognition task confirms that fact.

1 Introduction

Sequence modelling methods are applied in multiple areas. They are employed by computational biologists to model proteins [1]. The natural language processing community uses them to solve chunking or part-of-speech tagging tasks [2]. They are also applied in action recognition from video [3].

Probabilistic graphical models [4] are employed in sequence modelling. The generative Hidden Markov Model has been employed in many works. Multiple variations have been proposed to capture the peculiarities of different sequence modelling scenarios. Efficient exact and approximate algorithms exist to perform the associated inference tasks. Recently, discriminative sequence models such the Hidden Conditional Random Field (HCRF)[5] have emerged as a new alternative. They provide compact parametrizations and have higher predictive power. However, they still have reduced applicability and have not displaced generative models.

This work wants to foster the spread of discriminative sequence classifiers incorporating model and feature selection to the training algorithm of the HCRF. The Occams Razor principle of machine learning stands that a model should not be more complex than strictly required. Model and feature selection are two ways of implementing it, obtaining a more compact result. Model selection in the

context of the HCRF refers to the determination of the optimal number of hidden state variables, while feature selection refers to the selection of informative features in the input sequences while discarding uninformative ones.

1.1 Contributions

The contributions of this paper might be summarized as follows:

- A new training procedure for the HCRF incorporating model and feature selection.
- Experimental evidence showing that the proposed training algorithm performs better than the standard HCRF in a standard action sequence classification task.

1.2 Paper Organization

Paper is organized as follows: section 2 introduces the standard HCRF model; the proposed training procedure is presented on section 3; experimental evidence of the higher performance of the proposed method in a human action classification task is reported on section 4; finally, 5 resumes the contributions of this work and presents new research directions.

2 Hidden Conditional Random Fields

The HCRF [5] is an undirected graphical from the exponential family. It might be understood as an extension of the Conditional Random Field with hidden variables to model correlations among different observations. Multiple structured prediction tasks might be represented with HCRFs. This work assumes, without loss of generality, a sequence classification task.

Formally, the HCRF defines the conditional probability distribution of a discrete random variable $y \in \{y_1, \dots, y_N\}$ (a.k.a. sequence label) given a sequence of random variables $\mathbf{x} = x_1, \dots, x_T$ (a.k.a. observations) employing a set of auxiliary discrete hidden variables $\mathbf{h} = h_1, \dots, h_T$, $h_i \in \mathcal{H}$ not observed during training. These variables are introduced to model correlations among the observations in \mathbf{x} . In the case of sequence classification, these correlations correspond to the sequence dynamics. The conditional probability of the sequence label y and the hidden variable assignments \mathbf{h} given the sequence of observations \mathbf{x} is defined using the Hammersley-Clifford theorem of Markov Random Fields:

$$P(y, \mathbf{h} \mid \mathbf{x}, \theta) = \frac{e^{\Psi(y, \mathbf{h}, \mathbf{x}; \theta)}}{\sum_{y'} \sum_h e^{\Psi(y', \mathbf{h}, \mathbf{x}; \theta)}} \quad (1)$$

The conditional probability of the class label y given the observation sequence \mathbf{x} is obtained marginalizing over all the possible value assignments to hidden parts \mathbf{h} :

$$P(y \mid \mathbf{x}, \theta) = \frac{\sum_h e^{\Psi(y, \mathbf{h}, \mathbf{x}; \theta)}}{\sum_{y'} \sum_h e^{\Psi(y', \mathbf{h}, \mathbf{x}; \theta)}} \quad (2)$$

The potential function $\Psi(y, \mathbf{h}, \mathbf{x}; \theta)$ measures the compatibility of the input \mathbf{x} with the assignments to the hidden variables \mathbf{h} and the class label y . There are multiple possibilities about the form of this function. Here it is defined as:

$$\Psi(y, \mathbf{h}, \mathbf{x}; \theta) = \sum_{t=1}^T \phi(x_t) \alpha(h_t) + \sum_{t=1}^T \beta(h_t, y) + \sum_{t=1}^T \gamma(h_t, h_{t+1}, y) \quad (3)$$

where $\phi(x_t) \in \mathcal{R}^d$ is the feature vector associated with the observation x_t and $\theta = [\alpha \ \beta \ \gamma]$ is the vector of model parameters, indexed according to the values given to the hidden variables \mathbf{h} and label y . The first term, parametrized by $\alpha(h_t) \in \mathcal{R}^d$ measures the compatibility of the observation at instant x_t with the assignment to the hidden variable h_t . The second term measures the compatibility of the values given to the hidden parts h_t with the class label y and is parametrized by $\beta(y, h_i) \in \mathcal{R}$. Finally, the third term, parametrized by $\gamma(y, h_t, h_{t+1}) \in \mathcal{R}$ models sequence dynamics, measuring the compatibility of adjacent hidden variable assignments h_t and h_{t+1} with the class y .

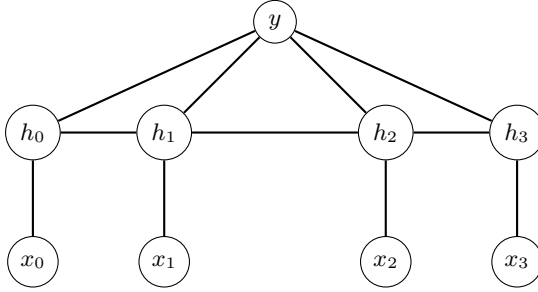


Fig. 1. Graphical model representing the structure of the HCRF induced by the function Ψ

The function Ψ induces the structure of the undirected graphical model defined by the HCRF. The structure of this graph can be observed on figure 1. Exact inference of the conditional probability distribution defined in equation 2 is possible, as the dependencies among the values given to the hidden variables \mathbf{h} form a chain. Efficient inference is achieved employing belief propagation [4].

2.1 Parameter Estimation

Optimal model parameters θ^* are estimated from a set of K training samples (\mathbf{x}^i, y^i) , $1 \leq i \leq K$, minimizing the L_2 regularized negative conditional log-likelihood function:

$$\theta^* = \arg \min_{\theta} L(\theta) = \arg \min_{\theta} - \sum_{i=1}^K \mathcal{L}(\mathbf{x}^i, y^i; \theta) + \lambda R(\theta). \quad (4)$$

The first term measures how model parameters are adjusted to predict each one of the K training samples, while the second term acts as a regularization prior over model parameters. The standard regularization employed in the HCRF is the Ridge regularizer, defined as $R(\theta) = \|\theta\|_2^2$, imposing a zero-mean gaussian prior on the values of θ to prevent overfitting. The parameter λ defines a trade-off between regularization and adjustment. A value of $\lambda = \frac{1}{2\sigma^2}$ is equivalent to a gaussian with variance σ^2 . The conditional log-likelihood function $\mathcal{L}(\mathbf{x}, y; \theta)$ is defined as:

$$\mathcal{L}(\mathbf{x}, y; \theta) = \log P(y | \mathbf{x}, \theta) = \log \left(\frac{\sum_{\mathbf{h}} e^{\Psi(y, \mathbf{h}, \mathbf{x}; \theta)}}{\sum_{y'} \sum_{\mathbf{h}} e^{\Psi(y', \mathbf{h}, \mathbf{x}; \theta)}} \right) \quad (5)$$

Due to the presence of the hidden variables \mathbf{h} , the objective function in equation 4 is non-convex [6]. However, a local optimum θ^* for the model parameter values might be obtained employing standard convex optimization techniques, as the function in 4 has a smooth gradient.

Different search strategies might be employed to find the optimal parameter values. Among them, the LBFGS quasi-newton method is the most popular [7], updating the descent direction with an approximation of the Hessian based on previous gradient estimations. Others have proposed to employ an online stochastic gradient descent algorithm [7], achieving a fast convergence rate but at the cost of obtaining a worst quality solution. In any case, the non-convexity of the objective function to optimize makes necessary to run the search multiple times from different starting points.

2.2 Limitations

The standard method to estimate HCRF optimal parameters leaves some open issues that are going to be discussed in order to motivate the proposal in subsequent section. These are:

- **How many hidden state variables employ?** $|\mathcal{H}|$ i.e., the number of different values that the hidden state variables in \mathbf{h} can take, should be specified *a priori*. If it is too small, the model is not enough expressive to capture the required correlations. However, if it is too big, noisy correlations are modelled and the result has a low predictive performance. Thus, it is necessary to adjust it to the right number. In practice, this is done employing cross-validation, evaluating the predictive performance for different choices and selecting the best. The non-concavity of the loss function in equation 4 complicates this process, as many trials should be made per choice to obtain a fair estimation of the optimality of each value. Thus, an efficient procedure is needed.
- **What happens if there are irrelevant features in the input sequences?** The L2 norm in equation 4 gives a non-zero weight to the parameters $\alpha(h_t)$ corresponding to irrelevant features. Thus, the result model does not have an optimal performance, as noise is incorporated to the inference

process. Thus, it is necessary to incorporate a method to select appropriate features from the input while discarding the irrelevant.

Other problem in the estimation of optimal HCRF parameters is how to adjust the trade-off between parameter fitting and regularization, i.e., what value give to λ in equation 4. This problem is shared by every regularized log-linear model. In practice, λ is adjusted employing cross-validation, needing to try different values until the one with the best performance is obtained. This adds another cross-validation dimension, as it should be already employed in the selection of the right number of hidden state values. The problem of estimating the right value for λ is out of the scope of this paper.

3 Model and Feature Selection in Hidden Conditional Random Fields

This section presents an overlapping group-L1 regularization strategy to estimate optimal parameters for the HCRF sequence classifier. As described in previous section, the components of the HCRF parameter vector θ are divided into three groups $\alpha(h_t)$, $\beta(h_t, y)$ and $\gamma(h_t, h_{t+1}, y)$, respectively indexed by the values of h_t , h_t and y and h_t, h_{t+1} and y . To obtain a model selection effect it is necessary to obtain zero values for all the parameters related to each unnecessary h . In a similar way, to perform feature selection it is necessary to obtain a zero value for all the parameters related to irrelevant input features.

Model and feature selection in log-linear models has been reported replacing L2 regularization of the objective function by L1 regularization[8]. However, L1 regularization is not enough to obtain model and feature selection in the HCRF as it only gives zero values to single variables and not to groups of them.

One way of obtaining zeros in groups of variables is employing overlapping group L1 regularization [9,10]. Be \mathcal{G} the power set of the parameter vector θ , and $G \subseteq \mathcal{G}$ an arbitrary subset of the power set. The overlapping group-L1 regularized training of the HCRF is given by the solution to the optimization problem:

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta) + \sum_{g \in G} \lambda_g \|\theta_g\|^2 \quad (6)$$

The overlapping group-L1 norm sums the L2 norm of the different groups defined in G . At the optimal, some of the groups will have a zero norm, as all the components from those groups will have become zero. Depending on the way the set G is defined, model selection, feature selection, both or even other advanced effects might be achieved:

- If $G \equiv G_{fs} = \cup_{d=1}^D \{\alpha(\cdot)_d\}$ feature selection is performed, as the L2 norm of the input features is penalized. A zero weight is expected for all the parameters corresponding to an input feature. Note that beta and gamma parameters are also regularized in order to prevent a big value on them, causing overfitting.

- If $G \equiv G_{ms} = \cup_{h=1}^{|\mathcal{H}|} \{\alpha(h) \cup \beta(h, \cdot) \cup \gamma(h, \cdot, \cdot) \cup \gamma(\cdot, h, \cdot)\}$ model selection is performed, as the L2 norm of the parameters corresponding to a hidden variable is minimized. A zero weight is expected to the parameters corresponding to non necessary hidden parts.
- If $G \equiv G_{fs} \cup G_{ms}$ model and feature selection are performed at the same time.

3.1 Optimization Algorithms

The convex optimization methods employed to estimate the optimal parameters of the standard HCRF are no longer valid. The new regularization term makes the objective function to optimize non-smooth. In particular, the gradient has a singularity at the points where a group gets a zero L2 norm. It is necessary to transform the problem into a smooth one before applying a gradient based method.

The unconstrained optimization problem in equation 6 might be reformulated into an equivalent constrained optimization problem as suggested by [11]:

$$\begin{aligned} \theta^* &= \min_{\theta} \mathcal{L}(\theta) + \sum_{g \in G} \lambda_g h_g \\ \text{s.t.} & \\ &\forall g \quad \|\theta_g\|_2 \leq h_g \end{aligned} \tag{7}$$

The overlapping group-L1 regularization term is replaced by a set of constraints, one for each group of variables in G . Each one of the constraints in the new optimization problem defines a norm cone of radius h_g , ensuring that the L2 norm of each group is smaller than h_g . A norm cone is a convex set, and the intersection of a set of convex sets is also a convex set [6]. Thus, the feasible region defined by the restrictions is convex. The norms of the different groups are added to the objective function. At the optimum the constraints are fulfilled with equality (it is trivial to probe that if they are not then it is not the optimal).

The objective function of the optimization problem in equation 7 is smooth, as the cause for the singularities has been removed. The estimation of the optimal parameters is made employing a gradient descent method, projecting the obtained values into the feasible set defined by the restrictions.

Dykstra’s algorithm [12] solves the problem of projecting a point $w_0 \in \mathcal{R}^k$ into the intersection of a set of convex sets $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_q$, alternately projecting the point into each set and removing the residual from the previous step.

To obtain the optimal parameter values different search methods have been proposed in [11]. Here the Projected Quasi-Newton (PQN) optimization method is employed. It builds a second-order approximation of the objective function around the current point to find the minimizing direction. The method avoids evaluating the objective function in the neighbourhood, assuming that computing the projections is cheaper than evaluating the objective function. Readers are referred to the original publication for further details on the method.

4 Experimental Evaluation

This section provides experimental evidence about the improvements that overlapping group-regularized training of HCRF models produces in their predictive power.

4.1 Experimental Setup

The system presented in figure 2 has been built to test the proposed method in a human action sequence classification task. The distance transform [13] is computed for each one of the human silhouettes extracted from the frames in the input sequence. A 3072 dimensional descriptor is obtained for each frame. The resulting sequence is introduced to the trained HCRF model to predict action class.

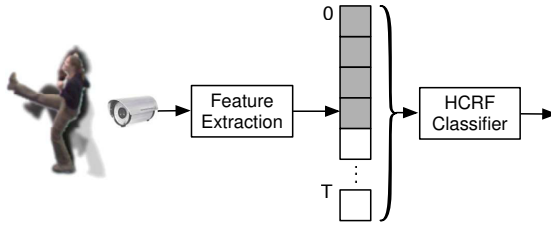


Fig. 2. Action Recognition Pipeline employed for evaluation

The models to be tested in order to evaluate the proposal are.

1. HCRF: The standard HCRF model as shown on section 2, employing L2 regularization. Optimal model parameters are obtained with LBFGS optimization.
2. MFS-HCRF: The Hidden Conditional Random Field trained with L1 group regularization to perform feature and model selection, as shown in section 3.

The predictive performance of these algorithms is going to be measured employing Weizmann dataset¹. It contains 10 different actions performed by 9 actors once, to give a total of 90 clips. Note that perfect classifications has been already reported for the dataset in [14]. However, the purpose of the experiments to be presented is to compare the performance of the presented algorithms in the task and not to try to provide a better way of performing Human Action Recognition.

The models are trained employing $|\mathcal{H}| = 20$ hidden parts, twice the number of action classes in Weizmann dataset. Iterative algorithms are applied until convergence. The non-convexity of the objective functions to be optimized forces to employ of a monte-carlo approach to evaluate each configuration to obtain fair results. Thus, each configuration in tested 30 times averaging the obtained results.

¹ <http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html>

4.2 Experiment I: Finding a Good Regularization Trade-Off

The first experiment to be conducted is to find for the different models a good value for the regularization parameter λ , providing a good equilibria between adaptation to the training data and regularization. The optimum is defined as the value minimizing the median negative log-likelihood obtained in the prediction of a test set. To this end sequences from Weizmann dataset are split in different subsets according to the actor. Sequences from actor 1 are employed as test set, while sequence from actors 2-9 are employed to train models.

Boxplots on figures 3(a) and 3(b) respectively show negative conditional log-likelihood values obtained for different values of λ for HCRF and MFS-HCRF. The negative log-likelihood values obtained for MFS-HCRF are smaller than the obtained for HCRF. Thus, the MFS-HCRF has a better predictive performance than the HCRF. Boxplots also show that the variance in negative log-likelihood values for the MFS-HCRF are slower than for the HCRF. This fact might be motivated by a softer objective function landscape, where local minima from the loss term of the objective function gets more penalized by the group regularization term.

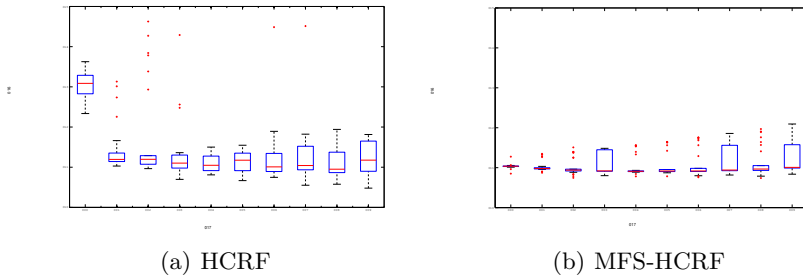


Fig. 3. Negative log-likelihood values achieved for different values of λ

4.3 Experiment II: Action Recognition Results

Previous experiment has shown that MFS-HCRF has higher predictive performance than HCRF for action sequences from a single actor. Now model performance is going to be measured in the prediction of the complete Weizmann dataset, measuring just predictive accuracy. This is done employing Leave One Actor Out Cross-Validation. Dataset is split again in different subsets according to the actor performing the sequence. The sequences from one actor are employed to measure the performance of models trained with the remaining actors. The process is repeated until every actor has been employed in the evaluation, joining the obtained results. The parameter λ is adjusted for the minimum value found in previous experiment.

Figures 4(a) and 4(b) present the confusion matrices respectively obtained for HCRF and MFS-HCRF. MFS-HCRF has a performance about a 2% higher than

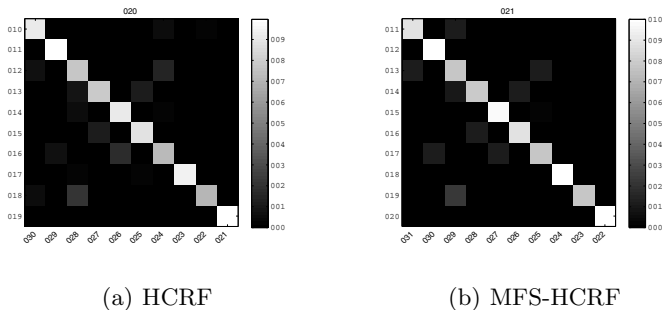


Fig. 4. Confusion matrices obtained for the different models in the prediction of Weizmann dataset

HCRF. Thus, the overlapping group-L1 regularized training of HCRF produces models with a higher predictive performance for the prediction of the action classes in Weizmann Dataset than those trained with standard L2 regularization.

5 Conclusions

This paper has presented a new training algorithm for the HCRF based on overlapping group-L1 regularization. Models trained with the proposed algorithm are more compact than the obtained by the standard algorithm, as model and feature selection is performed during training. Experiments have shown that the proposed algorithm recovers models with a higher predictive performance than the standard in an action recognition task.

Future works will validate the proposed method in other sequence classification tasks beyond human action recognition. The proposed algorithm might be adapted to provide model and feature selection in the estimation of optimal parameter values of other discriminative graphical models with hidden variables.

Acknowledgement. This work was supported in part by Projects MINECO TEC2012-37832-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485)

References

1. Krogh, A., Brown, M., Mian, I.S., Sjolander, K., Haussler, D.: Hidden markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology* 235, 1501–1531 (1994)
2. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: *Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing*, vol. 10, pp. 1–8. Association for Computational Linguistics (2002)

3. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden markov model. In: Proceedings of the 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1992, pp. 379–385 (1992)
4. Bishop, C., et al.: Pattern recognition and machine learning, vol. 4. Springer, New York (2006)
5. Quattoni, A., Wang, S., Morency, L.P., Collins, M., Darrell, T.: Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 1848–1853 (2007)
6. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press (2004)
7. Zhu, C., Byrd, R., Lu, P., Nocedal, J.: Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)* 23, 550–560 (1997)
8. Ng, A.: Feature selection, l_1 vs. l_2 regularization, and rotational invariance. In: Proceedings of the Twenty-First International Conference on Machine Learning, p. 78. ACM (2004)
9. Huang, J., Zhang, T.: The benefit of group sparsity. *The Annals of Statistics* 38, 1978–2004 (2010)
10. Szabó, Z., Póczos, B., Lorincz, A.: Online group-structured dictionary learning. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 2865–2872. IEEE (2011)
11. Schmidt, M.: Graphical model structure learning with l_1 -regularization. PhD thesis, University of British Columbia (2010)
12. Bauschke, H., Lewis, A.: Dykstras algorithm with bregman projections: A convergence proof. *Optimization* 48, 409–427 (2000)
13. Wang, L., Suter, D.: Visual learning and recognition of sequential data manifolds with applications to human movement analysis. *Computer Vision and Image Understanding* 110, 153–172 (2008)
14. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 2247–2253 (2007)

A First Approach to Deal with Imbalance in Multi-label Datasets

Francisco Charte¹, Antonio Rivera²,
María José del Jesus², and Francisco Herrera¹

¹ Dep. of Computer Science and Artificial Intelligence,
University of Granada, Granada, Spain

² Dep. of Computer Science, University of Jaén, Jaén, Spain
{fcharte,herrera}@ugr.es, {arivera,mjjesus}@ujaen.es
<http://simidat.ujaen.es>, <http://sci2s.ugr.es>

Abstract. The process of learning from imbalanced datasets has been deeply studied for binary and multi-class classification. This problem also affects to multi-label datasets. Actually, the imbalance level in multi-label datasets uses to be much larger than in binary or multi-class datasets. Notwithstanding, the proposals on how to measure and deal with imbalanced datasets in multi-label classification are scarce.

In this paper, we introduce two measures aimed to obtain information about the imbalance level in multi-label datasets. Furthermore, two pre-processing methods designed to reduce the imbalance level in multi-label datasets are proposed, and their effectiveness is validated experimentally. Finally, an analysis for determining when these methods have to be applied depending on the dataset characteristics is provided.

Keywords: Multi-label Classification, Imbalanced Datasets, Preprocessing, Measures.

1 Introduction

Classification is one of the most important tasks in the field of supervised learning. Multi-label classification (MLC) [1] is a generalization of binary and multi-class classification, as it does not impose an a priori limit to the number of elements that the set of outputs can hold. This type of classification is receiving significant attention lately, and it is being applied in fields such as text categorization [2] and music labeling [3], among others.

The data used for learning a classifier is often imbalanced, as the class labels assigned to each instance are not equally represented. This is a profoundly examined problem [4], but almost limited to binary datasets and to a lesser extent to multi-class datasets. That most multi-label datasets (MLDs) suffer from a large level of imbalance is a commonly accepted fact in the specialized literature [5], but there is a lack of measures to obtain information about it. In addition, and to the best of our knowledge, the proposals made until now to deal with imbalance in MLC have been focused in algorithmic adaptations of MLC algorithms [5–7], but none of them provides a general way of handling this problem.

In this paper two measures directed to determine the level of imbalance in MLDs are introduced, and two preprocessing methods aimed at reducing the imbalance in MLDs are proposed. The usefulness of the measures and effectiveness of the methods are proven experimentally, using different MLDs and MLC algorithms. The analysis of classification results provides a convenient guide in order to decide when an MLD suffers of imbalance and, therefore, could benefit from the preprocessing.

The rest of this paper is structured as follows: Section 2 briefly describes the MLC and the learning from imbalanced data problems. Section 3 introduces the imbalance problem in MLC, and presents the main proposals of the study which are the measures and preprocessing methods cited above. In Section 4 the experimental framework is described, and the results obtained are analyzed. Finally, the conclusions are presented in Section 5.

2 Preliminaries

2.1 Multi-label Classification

In many application domains [2,3,8] each data sample is associated with a set of labels, instead of only one class label as in binary and multi-class classification. Therefore, Y being the total set of labels in an MLD D , a multi-label classifier must produce as output a set $Z_i \subseteq Y$ with the predicted labels for the i -th sample. As each distinct label in Y could appear in Z_i , the total number of potential different combinations would be $2^{|Y|}$. Each one of these combinations is called a *labelset*. The same labelset can appear in several instances of D .

There are two main approaches [1] to accomplish an MLC task: data transformation and algorithm adaptation. The former aims to produce from an MLD a dataset or group of datasets which can be processed with traditional classifiers, while the latter has the objective of adapting existent classification algorithms in order to work with MLDs. Among the transformation methods the most popular are those based in the binarization of the MLD, such as Binary Relevance (BR) [9] and Ranking by Pairwise Comparison [10], and the Label Powerset (LP) [11] transformation, which produces a multi-class dataset from an MLD. In the algorithm adaptation approach there are proposals of multi-label C4.5 trees [12], algorithms based in nearest neighbors such as ML-kNN [13], multi-label neural networks [2,14], and multi-label SVMs [15], among others.

There are some specific measures to characterize MLDs, such as label cardinality *Card* and label density *Dens*. The former is the average number of active labels per sample in an MLD, while the latter is calculated as $Card/|Y|$ in order to obtain a dimensionless measure.

2.2 Classification with Imbalanced Data

The learning from imbalanced data problem is founded on the different distributions of class labels in the data [4], and it has been thoroughly studied in the

context of binary classification. In this context, the measurement of the imbalance level in a dataset is obtained as the ratio of the number of samples of the majority class and the number associated to the minority class, being known as *imbalance ratio* (IR) [16]. The higher the IR, the larger is the imbalance level. The difficulty in the learning process with this kind of data is due to the design of most classifiers, as their main goal is to reduce some global error rate [16]. This approach tends to penalize the classification of the minority classes.

In binary and multi-class classification the imbalance problem has been mainly faced using two different approaches: data preprocessing [17] and cost sensitive classification [18]. The former is based on the rebalancing of class distributions, either deleting instances of the most frequent class (undersampling) or adding new instances of the least frequent one (oversampling). Random undersampling (RUS) [19], random oversampling (ROS) and SMOTE [20] are among the most used preprocessing methods to equilibrate imbalanced datasets. The advantage of the preprocessing approach is that it can be applied as a general method to solve the imbalance problem, independently of the classification algorithms applied once the datasets have been preprocessed.

3 The Imbalance Problem in MLC

Most MLDs [21] have hundreds of labels, being each instance associated with a subset of them. Intuitively, it is easy to see that the more different labels exist, the more possibilities there are that some of them have a very low presence. In Figure 1, which represents the sample distribution per label of CAL500 dataset, this fact can be verified. However, as will be seen in Section 4, it is not straightforward

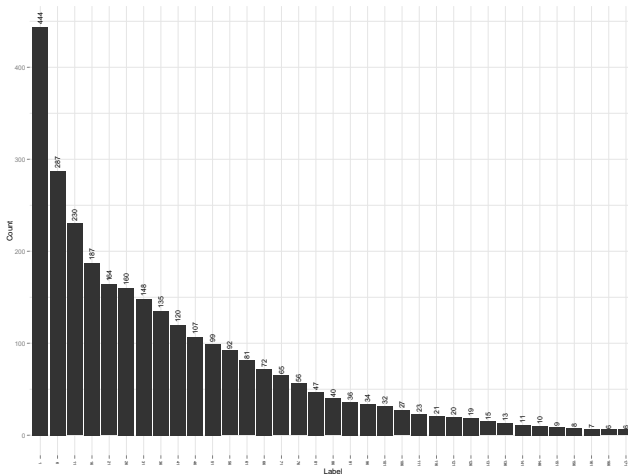


Fig. 1. Number of instances per label in CAL500 dataset

to infer the imbalance level from measures such as *Card* and *Dens*, which are the most widely used in the literature in order to characterize MLDs.

Many of the proposals made in the literature [5–7] for dealing with imbalanced datasets in MLC claim the imbalanced nature of MLDs, but none of them offer a procedure to measure it. Furthermore, most of these proposals aim to deal with the imbalance problem by means of algorithmic adaptations of MLC classifiers or the use of ensembles of classifiers. Therefore, there is a need for specific measures which can be used to obtain information about the imbalance level in MLDs, as well as some way able to face this problem while maintaining the use of the usual MLC algorithms.

3.1 Proposals on How to Measure the Imbalance Level in MLC

In traditional classification the imbalance level is measured taking into account only two classes: the majority class and the minority class. However, many MLDs have hundreds of labels, and several of them may have a very low presence. For that reason, it is important to define the level of imbalance in MLC considering not only two labels, but all of them. In this scenario, we propose the use of the following measures:

- *IRperLabel*: It is calculated for each label as the ratio between the majority and the considered labels, as shown in Equation 1. This value will be 1 for the most frequent label and a greater value for the rest. The higher *IRperLabel* is, the larger will be the imbalance level for the considered label.

$$IRperLabel(y) = \frac{\operatorname{argmax}_{y'=Y_1}^{|Y|} \sum_{i=1}^{|D|} h(y', Y_i)}{\sum_{i=1}^{|D|} h(y, Y_i)}, \quad h(y, Y_i) = \begin{cases} 1 & y \in Y_i \\ 0 & y \notin Y_i \end{cases}. \quad (1)$$

- *MeanIR*: This measure will offer a value which represents the average level of imbalance in an MLD, obtained as shown in Equation 2.

$$MeanIR = \frac{1}{|Y|} \sum_{y=Y_1}^{|Y|} (IRperLabel(y)). \quad (2)$$

- *CVIR*: This is the coefficient of variation of *IRperLabel*, and is calculated as shown in Equation 3. It will indicate if all labels suffer from a similar level of imbalance or, on the contrary, there are big differences in them. The higher is the *CVIR* the larger will be this difference.

$$CVIR = \frac{IRperLabel\sigma}{MeanIR}, \quad IRperLabel\sigma = \sqrt{\sum_{y=Y_1}^{|Y|} \frac{(IRperLabel(y) - MeanIR)^2}{|Y| - 1}}. \quad (3)$$

Table 1 shows the *MeanIR* and *CVIR* for the datasets used in the experimentation conducted for the present study. As we will see in the discussion on Section 4, these values would be enough to get a first glimpse to know the imbalance level in MLDs.

3.2 LP-RUS and LP-ROS: Random Undersampling and Oversampling for MLC

The existent undersampling and oversampling methods cannot be directly used in MLC, as they are designed to work with one output class label only. Furthermore, these methods assume that there are only one minority label and one majority label. Thus, an approach to preprocess MLDs, which have a set of labels as output and several of them could be considered minority/majority labels, is needed. In this paper we propose two methods aimed to undersample and oversample MLDs, called LP-RUS and LP-ROS. Both are based on the LP transformation method [11], which has been used in order to transform MLDs, in classification algorithms such as RAKEL [22] and HOMER [23], and also to complete other kinds of tasks, such as the stratified partitioning of MLDs [24]. Therefore, LP-RUS and LP-ROS will interpret each labelset as class identifier while preprocessing an MLD.

LP-RUS is a multi-label undersampling method that deletes random samples of majority labelsets, until the MLD D is reduced to a 75% of its original size. This method works as follows:

```

1: procedure LP-RUS( $D$ )
2:    $samplesToDelete \leftarrow |D| * 0.25$  ▷ 25% size reduction
3:   for  $i = 1 \rightarrow |labelsets|$  do ▷ Group samples according to their labelsets
4:      $labelSetBag_i \leftarrow samplesWithLabelset(i)$ 
5:   end for
6:   ▷ Calculate the average number of samples per labelset
7:    $meanSize \leftarrow 1/|labelsets| * \sum_{i=1}^{|labelsets|} |labelSetBag_i|$ 
8:   ▷ Obtain majority labels bags
9:   for each  $labelSetBag_i$  in  $labelSetBag$  do
10:    if  $|labelSetBag_i| > meanSize$  then
11:       $majBag_i \leftarrow labelSetBag_i$ 
12:    end if
13:  end for
14:   $meanReduction \leftarrow samplesToDelete/|majBag|$ 
15:   $majBag \leftarrow SortFromSmallestToLargest(majBag)$ 
16:  ▷ Calculate # of instances to delete and remove them
17:  for each  $majBag_i$  in  $majBag$  do
18:     $reductionBag_i \leftarrow \min(|majBag_i| - meanSize, meanReduction)$ 
19:     $remainder \leftarrow meanReduction - reductionBag_i$ 
20:     $distributeAmongBags_{j>i}(remainder)$ 
21:  for  $n = 1 \rightarrow reductionBag_i$  do

```

```

22:          $x \leftarrow \text{random}(1, |\text{majBag}_i|)$ 
23:          $\text{deleteSample}(\text{majBag}_i, x)$ 
24:     end for
25: end for
26: end procedure

```

The procedure described above aims to achieve a labelset representation in the MLD as close as possible to an uniform distribution. However, since a limit on the minimum dataset size has been established, a certain degree of imbalance among the labelsets could remain in the MLD. In any case, the imbalance level always will be lower than in the original dataset.

LP-ROS is a multi-label oversampling method that works cloning random samples of minority labelsets, until the size of the MLD is a 25% larger than the original. The procedure followed is analogous to the described above for LP-RUS. In this case, a collection of minority groups minBag_i with ($|\text{labelsetBag}_i| < \text{meanSize}$) is obtained, a $\text{meanIncrement} = \#\text{samplesGenerate}/\#\text{minBag}$ is calculated, and processing the minority groups from the largest to the smallest an individual increment for each minBag_i is determined. If a minBag_i reaches meanSize samples before incrementBag_i instances have been added, the excess is distributed among the others minBag . Therefore, the labelsets with a lower representation will be benefited from a bigger number of clones, aiming to adjust the labelset representation to an uniform distribution as in the case of LP-RUS.

4 Experimentation and Analysis

4.1 Experimental Framework

Four MLDs from the MULAN repository [21] were selected in order to test the proposed preprocessing methods. These are shown in Table 1, along with some measures which characterize them: number of attributes, samples, and labels, the average number of labels per sample, and the previously proposed measures related to the imbalance level. As can be seen, there are datasets with a variety of values in *Card* and *Dens*, as well as some big differences in the number of labels, attributes, samples, and the imbalance measures. The goal is to analyze how the proposed preprocessing methods work with MLDs which are not similar, but quite different.

Table 1. Characterization measures of datasets used in experimentation

Dataset	#Attributes	#Samples	#Labels	Card	Dens	MeanIR	CVIR
CAL500	68	502	174	26.0438	0.1497	20.5778	1.0871
Corel5k	499	5000	374	3.5220	0.0094	189.5676	1.5266
genbase	1186	662	27	1.2523	0.0464	37.3146	1.4494
scene	294	2407	6	1.0740	0.1790	1.2538	0.1222

The high *MeanIR* and *CVIR* values for Corel5k and genbase suggest that these MLDs are the most imbalanced, and therefore they could be the more benefited from the preprocessing. The CAL500 measurements also indicate a

certain level of imbalance, but *CVIR* is significantly lower than in the case of Corel5k and genbase, as is *MeanIR*. Finally, the values associated to scene denote its nature of well balanced MLD, being a dataset which does not need any preprocessing.

These datasets have been partitioned using a 5x2 folds cross validation scheme, and the training partitions have been preprocessed with LP-RUS and LP-ROS.

Regarding the MLC algorithms, the following methods were selected: BR-C4.5 [9], CLR [25], RAKEL [22], and IBLR-ML [26]. Each MLC algorithm was run over the base datasets, without any preprocessing, as well as using the datasets once they had been processed with LP-RUS and LP-ROS, respectively.

In the MLC field more than a dozen evaluation measures have been defined [1]. In this study to assess the influence of the preprocessing methods the following have been used: accuracy (Equation 4), precision (Equation 5) and recall (Equation 6). In these expressions Y_i is the set of real labels associated to the instance x_i , whereas $h(x_i)$ would be the set of labels predicted by the multi-label classifier.

$$Accuracy = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap h(x_i)|}{|Y_i \cup h(x_i)|} \quad (4)$$

$$Precision = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap h(x_i)|}{|h(x_i)|} \quad (5) \quad Recall = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap h(x_i)|}{|Y_i|} \quad (6)$$

Accuracy is a measure which assess the positive and negative predictive performance of the classifier, while precision is focused in the positive predictive performance only. Recall is a measure used often in conjunction with precision, and in this context will be useful to know the proportion of true active labels which has been predicted.

4.2 Results and Analysis

In Table 2 the accuracy for each configuration without preprocessing (noted as Base), with LP-RUS, and with LP-ROS are shown, and best values are highlighted in bold. It can be observed that LP-ROS always improves the results of Corel5k, and almost always in the case of genbase. These are the datasets with highest *MeanIR* and *CVIR* values. This implies that they are the most imbalanced in average, and that the differences in imbalance level among their samples is bigger than in the other MLDs, and that is something which LP-ROS is able to partially fix. The *MeanIR* of CAL500 is significantly lower, as is its *CVIR*. LP-ROS considerably improves the result of this MLD when processed with CLR, while losing narrowly with the others MLC algorithms. LP-RUS achieves some ties and slightly improves the result of CAL500/IBLR-ML. The scene dataset, characterized by a very low *MeanIR* and *CVIR* which denotes its nature of more balanced MLD, is the only one without any improvements.

Accuracy assesses positive and negative predictive performance. Table 3 shows the evaluation of results in terms of precision, a measure which only quantifies the positive predictive performance. This measure is generally used in conjunction

Table 2. Accuracy values on test sets

Dataset	Algorithm	Base	LP-RUS	LP-ROS
CAL-500	BR-J48	0.2135	0.2135	0.2060
CAL-500	RAkEL-BR	0.2135	0.2135	0.2060
CAL-500	CLR	0.1787	0.1787	0.2116
CAL-500	IBLR-ML	0.1922	0.1926	0.1900
Corel5k	BR-J48	0.0586	0.0480	0.0607
Corel5k	RAkEL-BR	0.0586	0.0480	0.0607
Corel5k	CLR	0.0360	0.0292	0.0446
Corel5k	IBLR-ML	0.0315	0.0235	0.0368
genbase	BR-J48	0.9842	0.9839	0.9844
genbase	RAkEL-BR	0.9842	0.9839	0.9844
genbase	CLR	0.9837	0.9812	0.9754
genbase	IBLR-ML	0.9790	0.9770	0.9804
scene	BR-J48	0.5318	0.5294	0.4648
scene	RAkEL-BR	0.5318	0.5294	0.4648
scene	CLR	0.5242	0.5194	0.4662
scene	IBLR-ML	0.6786	0.6683	0.6088

Table 3. Precision values on test sets

Dataset	Algorithm	Base	LP-RUS	LP-ROS
CAL-500	BR-J48	0.4398	0.4398	0.3448
CAL-500	RAkEL-BR	0.4398	0.4398	0.3448
CAL-500	CLR	0.6364	0.6364	0.5756
CAL-500	IBLR-ML	0.2859	0.2864	0.2776
Corel5k	BR-J48	0.3643	0.3638	0.1968
Corel5k	RAkEL-BR	0.3643	0.3638	0.1968
Corel5k	CLR	0.4620	0.4294	0.3624
Corel5k	IBLR-ML	0.0598	0.0451	0.0805
genbase	BR-J48	0.9947	0.9947	0.9939
genbase	RAkEL-BR	0.9947	0.9947	0.9939
genbase	CLR	0.9946	0.9946	0.9916
genbase	IBLR-ML	0.9899	0.9895	0.9922
scene	BR-J48	0.6752	0.6811	0.5989
scene	RAkEL-BR	0.6752	0.6811	0.5989
scene	CLR	0.6926	0.6998	0.6412
scene	IBLR-ML	0.8230	0.8164	0.7116

Table 4. Recall values on test sets

Dataset	Algorithm	Base	LP-RUS	LP-ROS
CAL-500	BR-J48	0.2964	0.2964	0.3446
CAL-500	RAkEL-BR	0.2964	0.2964	0.3446
CAL-500	CLR	0.2016	0.2016	0.2584
CAL-500	IBLR-ML	0.3722	0.3723	0.3782
Corel5k	BR-J48	0.0640	0.0516	0.0789
Corel5k	RAkEL-BR	0.0640	0.0516	0.0789
Corel5k	CLR	0.0378	0.0307	0.0491
Corel5k	IBLR-ML	0.0721	0.0690	0.0856
genbase	BR-J48	0.9896	0.9892	0.9904
genbase	RAkEL-BR	0.9896	0.9892	0.9904
genbase	CLR	0.9885	0.9858	0.9820
genbase	IBLR-ML	0.9867	0.9854	0.9867
scene	BR-J48	0.6295	0.6222	0.5826
scene	RAkEL-BR	0.6295	0.6222	0.5826
scene	CLR	0.6574	0.6454	0.6178
scene	IBLR-ML	0.6884	0.6809	0.6956

with recall (shown in Table 4), which is defined as the number of positive predictions versus real positives ratio.

Analyzing the effect of preprocessing methods with respect to precision and recall measures, the following can be observed: LP-ROS improves the recall in 12 of 16 configurations, but decreasing the precision. This means that LP-ROS produces a better coverage of labels which are present in the MLDs, but introducing false positives. On the contrary, LP-RUS improves the precision in some of the configurations, but the results with respect to recall are worse than the obtained by LP-ROS. This is due to the removing of false positives, but it also reduces the coverage of labels which should be present. When the preprocessing methods are applied to the scene dataset the results are not improved because of, as *MeanIR* and *CVIR* show, it could be considered as a balanced MLD.

From the analysis of these results, considering accuracy, precision and recall, it is possible to see that the LP-RUS preprocessing method, which reduces samples of the majority labelsets, obtains a slight improvement in precision but with significant costs. Intuitively, a labelset with a high representation in the MLD has

to be conformed by frequent labels, but the results show that frequent labelsets can include individual labels with low presence in other samples of the MLD. Thus, this preprocessing method reduces the presence of the most frequent labels, but also deletes samples in which not so frequent labels appear.

On the other hand, LP-ROS is a preprocessing method able to produce a general improvement, taking into account both positive and negative performance prediction (determined by means of accuracy, precision and recall measures), when applied over imbalanced MLDs. LP-ROS is a first approach to face with the imbalance problem for MLDs, and can be considered as a simple and efficient approach to improve the results of different MLC algorithms for imbalanced MLDs, i.e. with high *MeanIR* and *CVIR* values.

5 Conclusions

The classification with imbalanced datasets problem has been deeply studied, but almost limited until now to binary and multi-class contexts. In this paper two measures aimed to evaluate the imbalance level in MLDs, together with two preprocessing algorithms, have been proposed, and the experimentation made to validate them has been described. LP-RUS is a random undersampling algorithm, whereas LP-ROS does random oversampling, in both cases taking as class value the labelset assigned to each data instance.

The proposed measures can be used to assess the imbalance level, and being able to decide if a certain MLD could be benefited from the proposed preprocessing methods. We advanced in subsection 4.1, with the information offered by this measures, that Corel5k and genbase would be the most benefited MLDs, and that scene should not be preprocessed as it do not suffered from imbalance. The results discussed in subsection 4.2 have endorsed our hypothesis.

Among the two preprocessing algorithms proposed, LP-ROS obtains the best results considering different quality measures. We conclude that the multi-label oversampling accomplished by LP-ROS is able to improve classification results when it is applied to MLDs with large level of imbalance, such as Corel5k and genbase, whatever MLC algorithm is used.

Acknowledgments. F. Charte is supported by the Spanish Ministry of Education under the F.P.U. National Program (Ref. AP2010-0068). This paper is partially supported by the projects TIN2012-33856 and TIN2011-28488 of the Spanish Ministry of Science and Technology, FEDER funds, and the projects TIC-3928 and P10-TIC-6858 of the Andalusian Research Plan, FEDER funds.

References

1. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, ch. 34, pp. 667–685. Springer US, Boston (2010)

2. Zhang, M.-L.: Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. *IEEE Trans. Knowl. Data Eng.* 18(10), 1338–1351 (2006)
3. Wiczkowska, A., Synak, P., Raś, Z.: Multi-Label Classification of Emotions in Music. In: *Intel. Inf. Proces. and Web Mining*, ch. 30, vol. 35, pp. 307–315 (2006)
4. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.* 6(1), 1–6 (2004)
5. Tahir, M.A., Kittler, J., Bouridane, A.: Multilabel classification using heterogeneous ensemble of multi-label classifiers. *Pattern Recognit. Letters* 33(5), 513–523 (2012)
6. Tahir, M.A., Kittler, J., Yan, F.: Inverse random under sampling for class imbalance problem and its application to multi-label classification. *Pattern Recognit.* 45(10), 3738–3750 (2012)
7. He, J., Gu, H., Liu, W.: Imbalanced multi-modal multi-label learning for subcellular localization prediction of human proteins with both single and multiple sites. *PLoS One* 7(6), 7155 (2012)
8. Diplaris, S., Tsoumakas, G., Mitkas, P.A., Vlahavas, I.: Protein Classification with Multiple Algorithms. In: Bozaris, P., Houstis, E.N. (eds.) *PCI 2005*. LNCS, vol. 3746, pp. 448–456. Springer, Heidelberg (2005)
9. Godbole, S., Sarawagi, S.: Discriminative Methods for Multi-Labeled Classification. In: Dai, H., Srikant, R., Zhang, C. (eds.) *PAKDD 2004*. LNCS (LNAI), vol. 3056, pp. 22–30. Springer, Heidelberg (2004)
10. Hüllermeier, E., Fürnkranz, J., Cheng, W., Brinker, K.: Label ranking by learning pairwise preferences. *Artificial Intelligence* 172(16), 1897–1916 (2008)
11. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. *Pattern Recognit.* 37(9), 1757–1771 (2004)
12. Clare, A., King, R.D.: Knowledge discovery in multi-label phenotype data. In: Siebes, A., De Raedt, L. (eds.) *PKDD 2001*. LNCS (LNAI), vol. 2168, pp. 42–53. Springer, Heidelberg (2001)
13. Zhang, M., Zhou, Z.: ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognit.* 40(7), 2038–2048 (2007)
14. Zhang, M.-L.: ML-rbf: RBF Neural Networks for Multi-label Learning. *Neural Process. Lett.* 29, 61–74 (2009)
15. Elisseeff, A., Weston, J.: A Kernel Method for Multi-Labelled Classification. In: *Adv. Neural Inf. Processing Systems* 14, vol. 14, pp. 681–687. MIT Press (2001)
16. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. *Intell. Data Anal.* 6(5), 429–449 (2002)
17. Japkowicz, N.: Learning from imbalanced data sets: A comparison of various strategies, pp. 10–15. *AAAI Press* (2000)
18. Provost, F., Fawcett, T.: Robust classification for imprecise environments. *Machine Learning* 42, 203–231 (2001)
19. Kotsiantis, S.B., Pintelas, P.E.: Mixture of expert agents for handling imbalanced data sets. *Annals of Mathematics, Computing & Teleinformatics* 1, 46–55 (2003)
20. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* 16, 321–357 (2002)
21. Tsoumakas, G., Xioufis, E.S., Vilcek, J., Vlahavas, I.: MULAN multi-label dataset repository, <http://mulan.sourceforge.net/datasets.html>
22. Tsoumakas, G., Vlahavas, I.: Random k -labelsets: An ensemble method for multi-label classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) *ECML 2007*. LNCS (LNAI), vol. 4701, pp. 406–417. Springer, Heidelberg (2007)

23. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. In: Proc. ECML/PKDD Workshop on Mining Multidimensional Data, pp. 30–44 (2008)
24. Sechidis, K., Tsoumakas, G., Vlahavas, I.: On the stratification of multi-label data. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part III. LNCS, vol. 6913, pp. 145–158. Springer, Heidelberg (2011)
25. Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., Brinker, K.: Multilabel classification via calibrated label ranking. *Mach. Learn.* 73, 133–153 (2008)
26. Cheng, W., Hüllermeier, E.: Combining instance-based learning and logistic regression for multilabel classification. *Mach. Learn.* 76(2-3), 211–225 (2009)

Simulating a Collective Intelligence Approach to Student Team Formation

Juan M. Alberola, Elena del Val, Victor Sanchez-Anguix, and Vicente Julian

Departament de Sistemes Informàtics i Computació,
Universitat Politècnica de València,
Camí de Vera s/n. 46022, València. Spain
{jalberola, edelval, sanguix, vinglada}@dsic.upv.es

Abstract. Teamwork is now a critical competence in the higher education area, and it has become a critical task in educational and management environments. Unfortunately, looking for optimal or near optimal teams is a costly task for humans due to the exponential number of outcomes. For this reason, in this paper we present a computer-aided policy that facilitates the automatic generation of near optimal teams based on collective intelligence, coalition structure generation, and Bayesian learning. We carried out simulations in hypothetical classroom scenarios that show that the policy is capable of converging towards the optimal solution as long as students do not have great difficulties evaluating others.

1 Introduction

In the last few years, educational organizations have shown a growing interest in shifting towards teaching paradigms that promote teamwork. In fact, teamwork is now considered as a prominent general competence in the European Higher Education Area [12,15]. The inclusion of teamwork as a general competence responds to two main reasons: (i) it is strongly related to cooperative learning, a methodology that has been reported to enhance learning in the classroom [10,9]; (ii) nowadays, most successful business and engineering projects are carried out by small multidisciplinary teams [18,4]. Nevertheless, teamwork is a *double-edged sword* that can bring both positive and negative consequences [7], making the task of team formation a complex one. Several factors like personality, expertise, competitiveness, and human behavior can interfere with the performance of the team [13].

Therefore, it is of crucial importance to identify teams that can perform correctly. Several scholars have studied under what circumstances positive teamwork can emerge [16,2]. One of these studies is the well-known Belbin's role inventory [2]. Belbin identifies nine behavior patterns (i.e., roles) that are useful for teams: plants, resource investigators, coordinators, shapers, monitors, teamworkers, implementers, finishers, and specialists. As some studies suggest, teams usually benefit from having a varied mix of roles [2,8].

Team formation can also be a cognitive complex task. For instance, a class formed by 30 students can yield up to 767746 different teams of sizes 3, 4, 5 and 6¹, and the

¹ It corresponds to the sum of the number of combinations of size 3, 4, 5, and 6 ($\sum_{i=3}^6 \binom{30}{i}$).

number of different partitions of the class in disjoint teams grows exponentially with the number of possible teams. The problem is especially complicated if teachers want to find an optimal or near optimal team allocation. Professionals in education would certainly benefit from policies and tools that guide the team formation process.

In this paper, we simulate a computer-aided policy for forming disjoint teams of students based on collective intelligence, coalition formation, and Bayesian learning. More specifically, students grade other teammates based on Belbin's role inventory after the completion of each group activity. The information regarding the predominant roles of each student is then updated via Bayesian learning. Teams are formed following a coalition formation mechanism that employs the information inferred via the Bayesian learning process.

The remainder of the article is organized as follows. Section 2 presents the policy workflow. Section 3 describes the formalization of the policy and the Bayesian learning method. Section 4 shows an experimental evaluation in order to validate the policy. Section 5 details some previous works related to our proposal. Finally, Section 6 presents some concluding remarks.

2 Policy Overview

As stated, the proposed policy relies on collective intelligence, coalition formation, and Bayesian learning to form proper distributions of students teams. This policy is aided by a software application that prevents teachers from carrying out the costly task of dividing students into optimal or near optimal teams. The workflow of tasks followed by the policy can be observed in Figure 1. Next, we describe the workflow in more detail.

1. At the start of the course, there is no prior information regarding the natural preferences of each student for one of Belbin's roles. The policy starts with the first group activity of the course.
2. The teacher indicates to the computer tool that there is a new group activity. Then, students are divided into disjoint teams for the task at hand. As we will observe in Section 3, the problem of dividing students into optimal and disjoint teams is casted to a coalition structure generation algorithm.
3. After that, the activity is carried out.
4. Once the activity is finished, each student should log in the software application. Then, he/she is presented with a full description of Belbin's roles. At that point, the student should classify each teammate into a role. The information is then gathered by the application and stored in a database.
5. After the peer evaluation, the application updates the information for each student. Then, Bayesian learning is applied to determine which roles are more predominant for each student (Section 3.2).
6. The process continues (go to step 2) until the course has finished.

3 Policy Formalization

In this section, we formalize our policy for dividing students into disjoint teams. First, we describe how dividing students into optimal teams is equivalent to a coalition

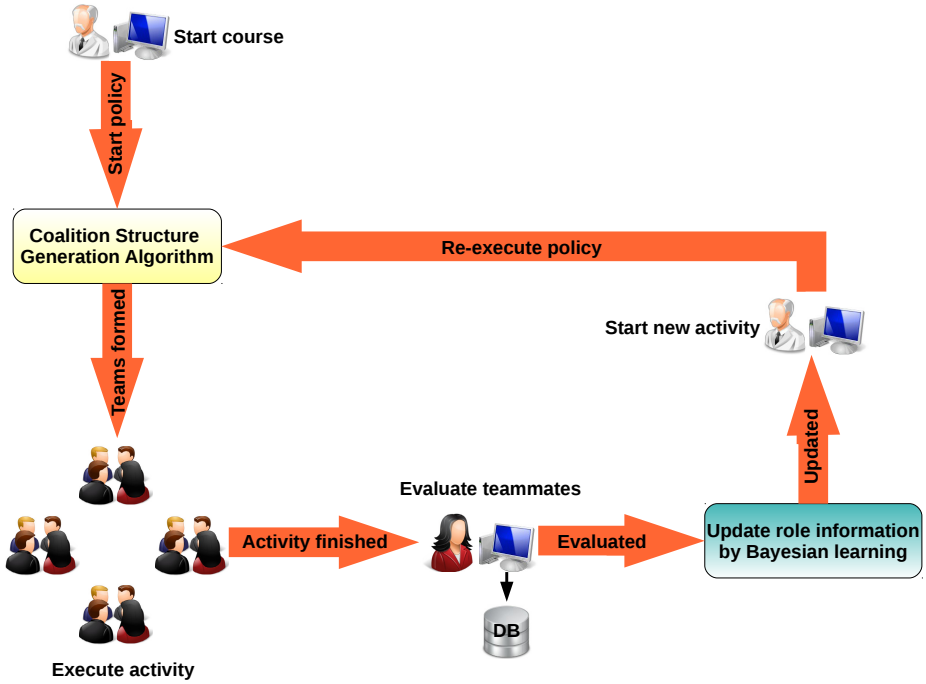


Fig. 1. An overview of the solution applied for grouping students into teams

structure generation problem. Then, we describe how Bayesian learning is employed in order to update the information of the classroom.

3.1 Student Team Formation as a Coalition Structure Generation Problem

Let $A = \{a_i, \dots, a_n\}$, be a set of students, and $R = \{r_1, r_2, \dots, r_m\}$ be the set of roles that the student can play (in our case it is the set of Belbin's roles), and let $role_i$ denote the true predominant role of a_i .

A subset $T \in A$ is called a *team*, and a *team structure* $S = \{T_1, T_2, \dots, T_k\}$ is a partition of disjoint teams such that $\bigcup_{T_j \in S} T_j = A$ and $S \in 2^A$. The goal of the application

is determining an optimal team structure for the classroom $argmax_{S \in 2^A} v(S)$, where $v(S)$ is an evaluation function for the team structure. In this study, we will consider that the quality of each team is independent of other teams. Hence, we can calculate the value of the team structure as $v(S) = \sum_{T_j \in S} v(T_j)$. The value of a team $v(T_j)$ can be calculated

attending to the predominant role that each student $a_i \in T_j$ has ($role(a_i)$). Let $|T_j| = k$ denote the size of the team and $\pi_j = \{r'_1, \dots, r'_k\}$ with $\forall r'_i \in R$ be a vector with the true predominant role of each team member. In that case, $v(T_j) = v(\pi_j)$. According to different studies [8], the team should benefit from having a balanced distributions of roles (i.e., one person per role). This score can be provided by an expert.

Unfortunately, it is not possible to accurately know the predominant role of each team member π_j and therefore $v(\pi_j)$ cannot be calculated with precision. However, it is possible for us to calculate an estimation of the value of the coalition given the history of evaluations H that is gathered from the students during the course. Let $\pi' = \{role_1 = r'_1, \dots, role_k = r'_k\}$ be a vector containing a set of hypotheses for the predominant roles of each team member, and Π be the set of all possible vectors of hypotheses for predominant roles of T_j . In that case, we can calculate the expected value of a team given the history of evaluations as:

$$\hat{v}(T_j|H) = \sum_{\pi' \in \Pi} p(\pi'|H) \times v(\pi') = \sum_{\pi' \in \Pi} v(\pi') \times \prod_{a_i \in T_j} p(role_i = r'_i|H) \quad (1)$$

where $p(\pi'|H)$ represents the probability for π' to be the real role distribution in T_j given the history of evaluations H . Each $p(\pi'|H)$ can be divided into its $p(role_i = r'_i|H)$ since we assume that the role of each student is conditionally independent given the history of evaluations. Therefore, our team formation problem at each iteration is casted out to one problem that follows the next expression:

$$argmax_{S \in 2^A} \sum_{T \in S} \hat{v}(T|H) \quad (2)$$

It turns out that partitioning a set students into disjoint teams while optimizing a social welfare function corresponds to the formalization of coalition structure generation problems. For our simulation experiments, we formalize the coalition structure generation problem as a linear programming problem [14] and solve it with the commercial software *ILOG CPLEX 12.5*².

3.2 Bayesian Learning

After every activity, students evaluate their peers by stating the most predominant role of each of his/her teammates. Then, new information becomes available regarding the most predominant role of each student and the history of evaluations H grows. Hence, at each iteration we can update information regarding the probability for an agent a_i to have r'_i as his/her most predominant role given the evaluation history $p(role_i = r'_i|H)$. We employ Bayesian learning for this matter :

$$p(role_i = r'_i|H) = \frac{p(H|role_i = r'_i) \times p(role_i = r'_i)}{\sum_{r \in R} p(H|role_i = r) \times p(role_i = r)} \quad (3)$$

where $p(H|role_i = r'_i)$ is the likelihood function and $p(role_i = r'_i)$ is the prior probability for the hypothesis. For the likelihood function, we can calculate it as $p(H|role_i = r'_i) = \frac{\#\{r'_i \in H_i\}}{|H_i|}$, where H_i denotes the peer evaluations about agent a_i , and $\#\{r'_i \in H_i\}$ indicates the number of times that r'_i appears as evaluation in H_i . As for the prior probability, we calculate it as $p(role_i = r'_i) = \frac{\#\{r'_i \in H\}}{|H|}$. Laplace smoothing [17] is employed to ensure that the likelihood for each role hypothesis can be calculated in the first iterations.

² <http://www.ibm.com/software/commerce/optimization/cplex-optimizer/>

4 Simulation Experiments

In this section, we simulate different classroom scenarios to study the behavior of the proposed policy throughout different group activities carried out in a course. First, we describe the general experimental setting, and then we describe the different experiments that we carried out and its results.

4.1 Experimental Setting

For the experiments, we simulate a classroom with $|A| = 20$ students and we employ all of the Belbin’s role except the specialist $|R| = 8$, since we consider that no student has specialized knowledge for the subject at hand.

The objective of policy is obtaining the optimal team structure. According to the Belbin’s taxonomy, this value is higher when a team is composed by heterogenous roles, i.e. when the predominant roles played by its teammates are different between themselves. As we stated in Equation 2, the expected value of a team structure is defined as the aggregation of the expected values of each individual team T given the history of evaluations H . As for the expected value of teams (see Equation 1), it depends on the aggregation of the probability of each possible role distribution π_j multiplied by the evaluation by an expert of such role distribution $v(\pi_j)$. For these simulations, we define $v(\pi_j) = \frac{MAX_v}{2^\gamma - 1}$, where $MAX_v = \frac{\#\{\text{different } r \in \pi_j\}}{|T_j|}$ is the number of different roles in π_j divided by the number of team members, and γ is the number of teammates playing repeated predominant roles. This way, we penalize those teams with less diversity.

In the policy, students classify other teammates according to Belbin’s taxonomy after the finalization of each activity. This process is simulated via Bernoulli distributions. Similarly to [1], each team member classifies other teammate into its corresponding predominant role according to a probability $\rho = [0..1]$. This probability is associated to the number of teammates with the same predominant role. In more detail, the probability of classifying a teammate into its predominant role is higher when no other teammate has the same predominant role. We set this probability to decrease with the number of teammates playing the same predominant role.

4.2 Results

In the first experiment we study scenarios where the distribution of predominant roles among students is uniform and students are grouped into teams of size 4. The simulation of each scenario is repeated 10 times to capture stochastic variations. In Figure 2 we show scenarios with different classifying probabilities $\rho = \{0.125, 0.25, 0.50, 0.75\}$. On the one hand, $\rho = 0.125$ represents a scenario where team members classify other team members randomly³ and $\rho = 0.25$ represents a scenario where students have difficulties to classify other teammates. On the other hand, $\rho = 0.5$ represents a scenario where there is an average difficulty for students to classify other team members, and $\rho = 0.75$ represents a scenario where team member easily classify other team members. The y-axis shows the average team structure value normalized between 0 and 1.

³ It is equivalent to a random classification since $\rho = \frac{1}{|R|} = \frac{1}{8}$.

We represent the average expected value of the solution found by the learning policy. Additionally, we also draw the value of the best team structure that can be found in the scenario.

As it can be observed, the real value keeps fluctuating when $\rho = 0.125$ since students provide no reliable information regarding other students. However, when $\rho = 0.25$, the real value found by the policy gradually converges towards the optimal solution. With just 5 iterations, the quality of the solution has improved a 10% over the initial solution. When $\rho = 0.5$ and $\rho = 0.75$, it can be observed that the real value converges in two iterations to the optimal value, approximately improving the initial solution a 30%. Since ρ is higher, the information provided by students can be considered reliable and the policy finds an optimal team structure faster. One important finding is that even with just 5 iterations, which represents a reasonable number of activities for a course, the policy is able to find reasonable good results even when students have difficulties evaluating other students.

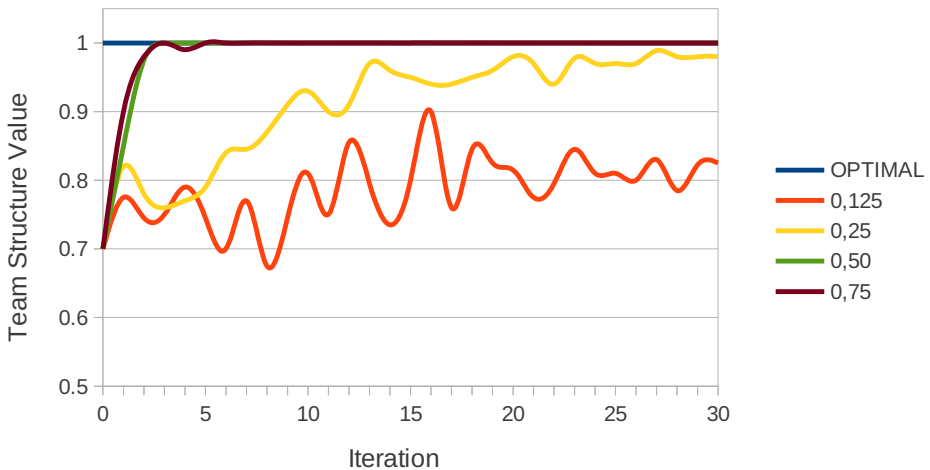


Fig. 2. Average team structure value found by the proposed policy when the role distribution among students is uniform

In the second experiment we test scenarios where the distribution of predominant roles is not uniform. More specifically, we set 3 out of the 8 roles to account for 50% of the student population. We tested scenarios where students have difficulties to evaluate other students $\rho = 0.25$ and scenarios where students have an average difficulty evaluating others $\rho = 0.5$. The rest of the parameters are set to the same value than in the previous experiment. We can observe the results for this experiment in Figure 3. For comparison purposes, we also include the analogous results for uniform distributions of roles among students. It can be observed that when roles are uniformly distributed, the policy converges more quickly towards the optimal solution than the analogous case in the non-uniform scenario. This can be explained due to the fact that the probability of putting students with the same role in a team is higher, then reducing the probability for students to classify other students correctly ρ . When students have difficulty to

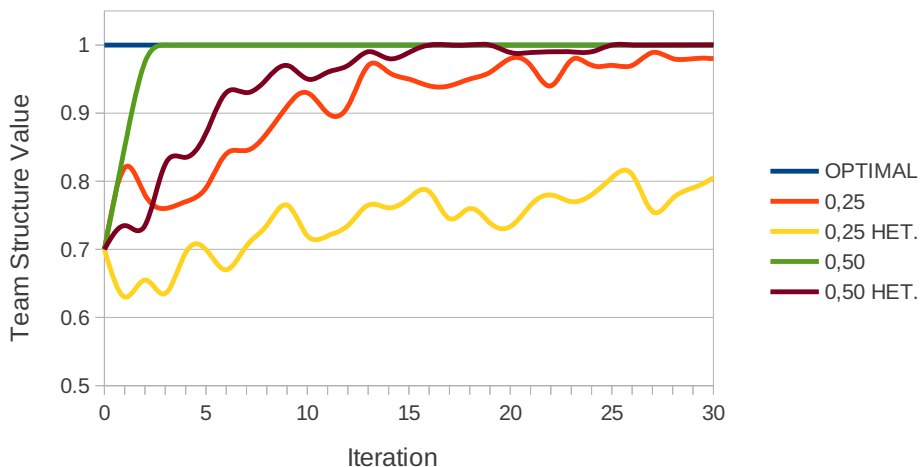


Fig. 3. Average team structure value found by the proposed policy when the role distribution among students is non-uniform

grade others ($\rho = 0.25$) the convergence is slow in the first iterations due to the fact that many students with the same role were placed together in the initial solutions, thus decreasing ρ . It is not until 15 iterations that the policy solution has improved approximately a 10% with respect to the initial solution. Nevertheless, the policy still converges towards the optimal solution even when students have an average difficulty to grade others $\rho = 0.5$. In just 5 iterations, it has been able to enhance the solution a 15% with respect to the initial solution. This result also suggests that the policy is applicable in a classroom as long as it is not highly difficult for students to evaluate others.

5 Related Work

When there is a large number of students and different grouping criteria, the task of forming collaborative learning teams to promote successful outputs is considered a NP-hard problem. Over the past years, several approaches have been proposed in order to deal with this goal.

The majority of the proposals try to create heterogeneous teams since there is a direct relationship between the performance of a team and the balance level among the roles. Christodouloupoulos et al. [3] present a web-based group formation tool that facilitates the creation of homogeneous and heterogeneous groups based on three criteria. This tool allows the instructor to manually modify the groups and allows the students to negotiate the grouping. The creation of homogeneous groups is based on a Fuzzy C-Means algorithm, and the creation of heterogeneous groups is based on a random selection algorithm. The tool also provides an option to negotiate the teams proposed with the students. However, this negotiation consists on a direct interaction with the teacher. Other approaches use bio-inspired algorithms. Graf et al. [5] present an Ant Colony Optimization approach that provides heterogeneous groups based on personal

Table 1. Comparison of approaches that deal with the problem of forming collaborative learning teams. The features considered are: the initial information available about the students, the algorithm used, the inclusion of feedback in the coalition formation process, population in each cluster (heterogeneous or homogeneous), and number of students used in the experiments.

	initial info.	algorithm	feedback	clusters	population
[3]	3 attributes	Fuzzy C-mean random selection	negotiation	het./hom	18
[5]	personal traits	ant colony optimization	-	het.	500
[19]	<i>thinking styles</i> (questionaries)	genetic alg.	final students satisfaction	het	66
[20]	personal knowledge social network	genetic alg.	-	het.	45
[11]	understanding level interest of students	particle swarm optimization	-	hom	15-2000
[21]	self-evaluation of roles	crowding evolutionary alg.	-	het.	18-3000

traits of students. The groups are formed by four students with low, average, and high students scores. The algorithm tries to maximize the diversity of the group while keeping a similar degree of heterogeneity of all the groups.

Genetic or evolutionary algorithms are also commonly used to solve the NP-hard problem of forming collaborative learning teams. Wang et al. [19] present an approach for automatic team formation based on *thinking styles* [6] that determines the features of the students. They consider an heterogeneous group formation. The algorithm translates the features of the students into points in a two-dimensional space and then, they are classified into categories. The algorithm uses a genetic algorithm to create the optimal group formation based on the categories of the students. The experiments consider 66 students and groups of 3 people. At the end of the process, the coalitions are evaluated by the students through a questionnaire. However, this evaluation is not used as feedback to improve the group formation. Lin et al. [11] present a system to assist instructors to form collaborative learning groups that provide outcomes to all their members. The algorithm considers two criteria: information about understanding levels and interests of the students. Particle swarm optimization is used in the group composition algorithm to deal with the complexity of the problem. The main drawback of this proposal is that the groups consist of homogeneous students. Yannibelli and Amandi [21] propose a crowding evolutionary algorithm to deal with the complexity of the problem of creating collaborative learning teams. The algorithm balances the roles of the members and the number of members of each group. Belbin's roles [2] are considered in the experiments and, in order to assign a role to each student they initially use the Team Role Self-Perception Inventory.

The problem of team formation is also present in the context of human resource management. Wi et al [20] present a framework to deal with the team formation in R&D-oriented institutes. The authors propose a genetic algorithm that uses a fuzzy model to take into consideration information about the candidates related to their knowledge and

expertise about certain topics related to certain project. Moreover, the algorithm considers information about the position of the candidates in a social network in order to see their suitability for project management positions.

Our proposal for collaborative learning teams formations improves previous approaches in several ways. First, our proposal does not have previous information about the abilities, attributes, or roles played by the students. The only previous information is the set of roles that could appear in a team. Using these roles, team members can provide an estimation about the roles played by other team members. Second, our proposal provides a more reliable role assignment since it considers the opinion of other members instead of a personal evaluation. Finally, in each iteration of the algorithm, the solution is improved with the feedback received from direct interactions among students in each team.

6 Conclusions

In this paper we have presented a computer-aided policy for generating teams of students a classroom. The policy is based on Belbin's role taxonomy [2], collective intelligence, coalition structure generation algorithms, and Bayesian learning. After the execution of a class group class activity, students classify other teammates according to Belbin's role taxonomy. Then, the information regarding the predominant role of each student is updated via Bayesian learning. This information is then used by the coalition structure generation algorithm to calculate the next team structure. The simulations have shown that, as long as students do not have great difficulties classifying others, the policy is capable of improving the quality of team structures in a few iterations and gradually converging towards the optimal solution.

We simulated different scenarios in order to test different environmental conditions. The results are encouraging enough to continue this research. As a future work, we plan to extend the experiments in order to consider large populations of students and environmental conditions, such as scenarios where some roles are more important than others. In addition, we also intend to study whether or not the inclusion of more attributes in the classification problem can improve the performance of the policy.

Acknowledgements. This work is supported by TIN2011-27652-C03-01 and TIN2012-36586-C03-01 of the Spanish government and FPU AP2008-00601 granted to Elena del Val.

References

1. Alberola, J.M., Julian, V., Garcia-Fornes, A.: Multidimensional Adaptation in MAS Organizations. *IEEE Trans. Syst. Man Cybern. Part B-Cybern.*, 1–12 (2012)
2. Belbin, R.M.: *Team roles at work*. Routledge (2010)
3. Christodouloupoulos, C.E., Papanikolaou, K.A.: A group formation tool in an e-learning context. In: *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, vol. 2, pp. 117–123 (2007)

4. De Vries, M.F.: High-performance teams: Lessons from the pygmies. *Organizational Dynamics* 27(3), 66–77 (2000)
5. Graf, S., Bekele, R.: Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006*. LNCS, vol. 4053, pp. 217–226. Springer, Heidelberg (2006)
6. Grigorenko, E., Sternberg, R.: Styles of thinking, abilities, and academic performance. *Exceptional Children* 63(3), 295–312 (1997)
7. Hansen, R.S.: Benefits and problems with student teams: Suggestions for improving team projects. *Journal of Education for Business* 82(1), 11–19 (2006)
8. Higgs, M., Pewina, U., Ploch, J.: Influence of team composition and task complexity on team performance. *Team Performance Management* 11(7/8), 227–250 (2005)
9. Johnson, D.W., Johnson, R.T.: An educational psychology success story: Social interdependence theory and cooperative learning. *Educational Researcher* 38(5), 365–379 (2009)
10. Johnson, D.W., Johnson, R.T., Smith, K.: The state of cooperative learning in postsecondary and professional settings. *Educational Psychology Review* 19(1), 15–29 (2007)
11. Lin, Y.-T., Huang, Y.-M., Cheng, S.-C.: An automatic group composition system for composing collaborative learning groups using enhanced particle swarm optimization. *Computers & Education* 55(4), 1483–1493 (2010)
12. Maffioli, F., Augusti, G.: Tuning engineering education into the european higher education orchestra. *European Journal of Engineering Education* 28(3), 251–273 (2003)
13. Morgeson, F.P., Reider, M.H., Campion, M.A.: Selecting individuals in team settings: The importance of social skills, personality characteristics, and teamwork knowledge. *Personnel Psychology* 58(3), 583–611 (2005)
14. Ohta, N., Conitzer, V., Ichimura, R., Sakurai, Y., Iwasaki, A., Yokoo, M.: Coalition structure generation utilizing compact characteristic function representations. In: Gent, I.P. (ed.) *CP 2009*. LNCS, vol. 5732, pp. 623–638. Springer, Heidelberg (2009)
15. Pajares, S., Sanchez-Anguix, V., Torreño, A., Esparcia, S.: A novel teaching-learning strategy for teamwork based on agreement technologies. *IJCA Proceedings on Design and Evaluation of Digital Content for Education (DEDCE)* (1), 21–30 (2011)
16. Ratcheva, V., Vyakarnam, S.: Exploring team formation processes in virtual partnerships. *Integrated Manufacturing Systems* 12, 512–523 (2001)
17. Russell, S.J., Norvig, P.: *Artificial Intelligence: A Modern Approach*. Prentice Hall (2010)
18. Tarricone, P., Luca, J.: Employees, teamwork and social interdependence—a formula for successful business? *Team Performance Management* 8(3/4), 54–59 (2002)
19. Wang, D.-Y., Lin, S.S., Sun, C.-T.: Diana: A computer-supported heterogeneous grouping system for teachers to conduct successful small learning groups. *Computers in Human Behavior* 23(4), 1997–2010 (2007)
20. Wi, H., Oh, S., Mun, J., Jung, M.: A team formation model based on knowledge and collaboration. *Expert Systems with Applications* 36(5), 9121–9134 (2009)
21. Yannibelli, V., Amandi, A.: A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context. *Expert Systems with Applications* 39(10), 8584–8592 (2012)

A Counting-Based Heuristic for ILP-Based Concept Discovery Systems

Alev Mutlu¹, Pınar Karagoz¹, and Yusuf Kavurucu²

¹ Department of Computer Engineering
Middle East Technical University, Ankara, Turkey
{mutlu,karagoz}@ceng.metu.edu.tr

² Turkish Naval Academy
ykavurucu@dho.edu.tr

Abstract. Concept discovery systems are concerned with learning definitions of a specific relation in terms of other relations provided as background knowledge. Although such systems have a history of more than 20 years and successful applications in various domains, they are still vulnerable to scalability and efficiency issues —mainly due to large search spaces they build. In this study we propose a heuristic to select a target instance that will lead to smaller search space without sacrificing the accuracy. The proposed heuristic is based on counting the occurrences of constants in the target relation. To evaluate the heuristic, it is implemented as an extension to the concept discovery system called C^2D . The experimental results show that the modified version of C^2D builds smaller search space and performs better in terms of running time without any decrease in coverage in comparison to the one without extension.

Keywords: Inductive Logic Programming, Concept Discovery, Search Space, Counting.

1 Introduction

Inductive Logic Programming (ILP) [1] is concerned with inducing general patterns that are valid for a given set of facts. Concept discovery [2], a research area which can be considered as a subproblem of ILP, deals with inducing patterns that are valid for a specific relation, called *target relation*, in terms of other relations provided, called *background knowledge*. Such systems usually employ first order logic as the representation language, and induce the target relation definitions, called *concept descriptors*, in the form of Horn clauses.

Although such systems have a history of more than 20 years and promising applications in several domains, they still sustain scalability and efficiency issues. These issues are generally due to the evaluation of the large search spaces they build. Generally speaking, ILP-based concept discovery systems start with an initial hypothesis set and refine it to find complete and consistent concept descriptors. The size of the search space is related to the size of the initial hypothesis set as the refinement of the search space is achieved by specializing

and generalizing the current search space. To form the initial search space, systems employ heuristics such as relative least general generalization (*rlgg*) [3], randomised restarted search [4], simulated annealing [5].

In this work, we propose a heuristic to select a target instance which generates a smaller initial hypothesis set and subsequently leads to a smaller search space. Our intuition is that, a target instance whose arguments appear less frequently compared to others should lead to smaller initial search space. The proposed heuristic is based on counting the frequencies of the constants that constitute the target instance set. To realize the heuristic, we count the frequencies of the constants that appear in the target instances and based on these values we calculate global frequency value of the target instances. The global frequency value of a target instance is equal to the summation of frequencies of the its arguments. Our heuristic considers that the target instance with the smallest global frequency value will lead to a smaller search space compared to the one of with a higher global frequency value.

In order to analyze the performance of the proposed heuristic, we implemented it as extension to an ILP-based concept discovery system called C^2D [6]. It is a hybrid concept discovery system that employs inverse resolution operator to generalize the concept descriptors, and an Apriori-based operator specialization operator. Experiments are conducted on several benchmark data sets. The experimental results show that the proposed heuristic builds smaller search spaces in comparison to starting C^2D with a randomly chosen target instance.

This paper is organized as follows. In Section 2, we briefly introduce the problem of concept discovery and provide overview of initial search space formation process of several ILP-based concept discovery systems. In Section 3, we introduce the C^2D system. In Section 4, we introduce our heuristic. In Section 5, we discuss the experimental results. We conclude the paper and list some future directions in Section 6.

2 Concept Discovery

Concept discovery is a predictive ILP problem. Concept discovery systems input a set of target instances, a set of background knowledge, some user defined quality metrics such as maximum concept descriptor length and minimum support. Such systems aim to output *complete* and *consistent* concept descriptors that qualify the user provided quality metrics. In context of concept discovery, a hypothesis is called complete if it models every positive target instance, and consistent if it covers no negative target instance. As such systems usually work on noisy and incomplete data, completeness and consistency principles are relaxed to cover as many positive target instances as possible and as few negative target instances as possible, respectively. Concept discovery systems usually employ first order logic as the representation framework for the input data, and output the concept descriptors in the form of Horn clauses.

Initial implementations of concept discovery systems were generally concentrated on concepts and techniques. Once their applicability on numerous domains [7–10] were proven to be successful, efforts that aim to improve their

scalability and efficiency arose. Such attempts include parallelization [11], memoization [12], and optimization of search space evaluation queries [13].

Another direction in coping with scalability and efficiency issues is the introduction of heuristics to shrink the size of the search space or to guide the search. Pruning the search space by means of introduction of mode declarations [14] and data integrity constraints [15] were proposed. Although such constraints greatly reduce the size of the search space, they are usually hard to define and may require expertise.

To guide the search, heuristics such as randomised restarted search [4], simulated annealing [5], gini index [16], and relative least general generalization [3] were proposed. In randomised restarts approaches searches are restarted when the search exhibits slower decay than expected. Simulated annealing approaches allow exploration of the search space more efficiently by allowing the specialization and generalization operators at the same time. Gini index and several other functions based on entropy allow to choose the best promising concept descriptor in the search space.

The heuristic we propose in this work differs from the ones mentioned above as it aims to choose a target instance which seems to produce a smaller initial search space. It is based on counting the number of appearances, i.e. frequencies, of constants that form the target instances in the background data and assigning global frequency to target instances based on arguments' frequencies. The proposed heuristic is cost efficient as its execution only once in the very beginning of the concept discovery process is enough as the background data is never altered.

3 The C^2D System

C^2D is a predictive ILP-based concept discovery system that employs association rule mining concepts and techniques. It employs absorption operator of inverse resolution for generalization and Apriori-based operator based on confidence for specialization. It inputs minimum support, minimum confidence and maximum rule length values to put constraints on the quality of the induced concept descriptors. C^2D distinguishes from state of the art concept discovery systems as it relaxes the requirement for mode declaration, negative examples, and discarding the aggregate predicates.

The system is composed of the following main functions:

- 1) Generalization: In this step the most general two literal concept descriptors are constructed. For this aim, C^2D selects the first uncovered target instance from the target instances set together with facts related to it from the background data. To build the most general two literal generalizations of the target instances, firstly, C^2D builds definite clauses with the target example as their heads and background relation as their bodies. Then, it either substitutes the constants with variables or keep them as constant if their occurrence in the background data is above some user defined threshold.

- 2) Specialization: In this step candidate concept descriptors of length l are specialized to build the candidate concept descriptors of length $l+1$. To build the specializations of a candidate concept descriptor it is compared to every other candidate concept descriptor in the search lattice and is unified with those that it differs with exactly one body literal.
- 3) Evaluation: In this support and confidence values of candidate concept descriptors are calculated. For this, they are translated into SQL queries which in turn are run against the background data.
- 4) Filtering: This step is concerned with the pruning the search space. A candidate concept descriptor is pruned if it does not satisfy the minimum confidence and support values or violates the data integrity constraints.
- 5) Covering: Once some concept descriptors are found that satisfy the minimum confidence and support values, target instances explained by them are marked as covered. If the number of the uncovered target instances are above the user defined threshold the systems goes to the *Generalization* step to discover concept descriptors to explain the reaming target instances.

To find concept descriptors, C^2D handles the uncovered target instances in turn. Initially all target instances are uncovered, and once some concept descriptors that satisfy the quality metrics are found, instances explained by them are marked as covered. In order to find concept descriptors to explain the remaining target instances, C^2D picks the first yet uncovered target instance and restarts the induction process.

4 The Proposed Heuristic

4.1 Motivation

As described in Section 2, concept discovery systems start with building an initial hypothesis set and grow the search space accordingly. Such search spaces generally grow fast as the operators refine each hypotheses in every possible way to find complete and consistent concept descriptors. A concept discovery system that starts with a larger initial hypothesis set is more likely to end up with a larger search space compared to the one which starts with a smaller initial hypothesis set.

Target instances in concept discovery systems are directly or indirectly related to the background data, which means their arguments are spread across background relations. In order for the induced concept descriptors to be complete and consistent, they should include the background relations that contain the target instance arguments. Assume that two of the constants that appear in the target instance set are Arg_1 and Arg_2 . Further assume that Arg_1 appears almost in every background relation, and Arg_2 appears only in relations R_1 , R_2 , R_3 . In such a scenario:

- a) As Arg_1 appears in many background relations, initial hypotheses derived from a target instances that contains Arg_1 will lead to a larger initial hypothesis set.

- b) In order the induced concept descriptors to be complete and consistent, they should include R_1 , R_2 and R_3 , not necessarily all of them together, as only these three relations are related to Arg_2 .

Based on these discussions, we propose to pick the target instances whose sum of argument frequencies is minimum as the seed target instance to build the initial hypothesis set from.

4.2 The Heuristic

As indicated in Section 4.1, the proposed heuristic involves counting the frequencies of constants that constitute the target instances and assigning a global frequency values to each target instance. Realization of the heuristic requires the implementation of the Algorithm 1.

Algorithm 1. The proposed heuristic

Require: E : target instances, B : Background knowledge

Ensure: E' : Target instances sorted in ascending order based on their global frequencies

```

1: for each  $e_i$  in  $E$  do
2:   for each argument  $a_j$  in  $e_i$  do
3:     if (notCountedBefore( $a_j$ )) then
4:       counts.insert( $a_j$ , count( $a_j$ , B))
5:     else
6:        $A_j =$  counts.find( $a_j$ )
7:     end if
8:      $e_i = \sum(A_1, A_2, \dots, A_j)$ 
9:   end for
10: end for
11:  $E' =$  SortInAscendingOrder( $E$ )
12: return  $E'$ 

```

The algorithm considers each argument of each target instance in turn. Before counting the frequency of an argument, it is first searched in a look-up table called *counts*. As target instances may have some constants in common, each distinct constant is counted once at its first appearance in the target instance set, and thereafter its frequency value is retrieved from the look-up table. Once frequency of each argument is calculated, global frequency of a fact is calculated by summing up the frequencies of the constants that take part of that target instance.

The *count* function given at line 4 counts the appearances of the constant in the background data, and the function *SortInAscendingOrder* given on line 11 sorts the target instances in ascending order based on their global frequency values.

Such a heuristic has two rationales:

- a) As it selects a target instance that generates a small initial hypothesis set, the search space that will be built is likely to be small.
- b) As this heuristic picks a target instance whose arguments appear less frequently, it is likely that it will consider the background relations which are likely to have high coverage as more frequent constants also appear in that particular background relations.

4.3 Applicability to Other Systems

The proposed heuristic may be embodied into other systems. The proposed heuristic can be incorporated in *rlgg* to further optimize the system when several pairs of target instances have the same coverage value. Application of the proposed heuristic to gini index is similar to its applicability to *rlgg*. The heuristic may be of help to choose the most promising target instance when several target instances have the same gini index value. Application of the proposed heuristic to the randomised restarts method is also possible. The heuristic may be utilized to select the next target instance which is likely to produce a smaller search once the current one bears the search limits.

As these examples suggest, the proposed heuristic may be embodied into several systems as helper to choose among several target instances when they have the same or close quality metrics, instead of picking one of them randomly.

5 Experiments

5.1 Data set and the Experimental Setting

In order to evaluate the performance of the proposed heuristic, we conducted a set of experiments on Large Family¹ [17], Mesh [18], PTE [19], and Same Generation [6] data sets.

In Table 1, we provide some statistical information regarding the data sets. In each row, the table provides information about the number of target instances, number of background facts, the minimum and the maximum frequencies of the constants that take part in the target instance set. The last column indicates the average frequencies of the constants per background relation. Except for the Same Generation and the PTE data set, average frequency of the constants is around 1.7 regardless of the maximum and minimum frequencies. Therefore, we can deduce that the high frequency of a constant does not mean that it is heavily contained in certain background relations, rather means that its occurrence is scattered over many background relations with less coverage. In the experiments, other than Mesh, minimum support is set to 0.3, minimum confidence is set 0.7, and maximum concept descriptor length is set to 3. For the Mesh data set the minimum support is set to 0.1 while the other parameters remain unchanged.

¹ This data consists of 12 kinship relations. While conducting experiments on this data set, we picked each relation, in turn, as a target relation and the others as background knowledge.

Table 1. Statistic on the data set

Relation Name	# Target Instances	# Background Instances	Minimum Frequency	Maximum Frequency	Appearance per Table
Aunt	82	662	4	24	1.7
Brother	59	685	4	33	1.8
Daughter	54	690	5	33	1.8
Father	60	684	4	30	1.8
Husband	25	719	3	28	1.9
Mother	60	8684	3	26	1.8
Nephew	82	662	5	27	1.7
Niece	92	652	7	28	1.7
Sister	47	697	8	34	1.8
Son	66	678	3	30	1.7
Uncle	92	652	7	26	1.7
Wife	25	719	3	32	1.94
Same Generation	752	408	14	24	7.3
Mesh	223	1749	5	12	1.1
PTE	162	23850	34	304	7.9

5.2 Experimental Results

To evaluate the proposed heuristic, we conducted two types of experiments. In the first experiment, we sorted the target instances in ascending order based on their global frequencies. In the second experiment we listed the target instances randomly. Then we picked the target instances from these sorted and unsorted lists. As the order of the instances effect the induction process and the size of the search space, we conducted the second experiment 5 times for each data set – at each experiment the target instances were randomly resorted. In Table 2 we list the experimental results. Results listed under Random Order are the average of five runs.

The proposed heuristic is compared to starting the concept discovery process with randomly selected target instances by means of the number of queries executed to evaluate the search space, and the duration of the concept induction process.

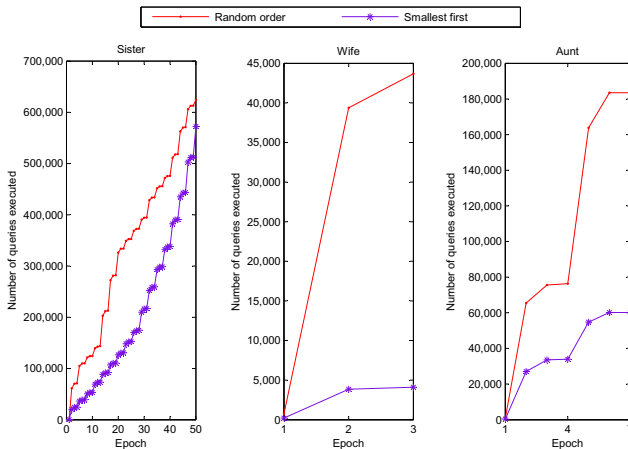
In Table 2, under the *Improvement* column we list the drop in the number of queries executed to evaluate the search space and the speedup. Although speedup is metric to evaluate the performance of parallel algorithms, it is also widely employed to compare performance of different implementations of the same serial algorithm. Speedup is calculated by dividing the running time of the original implementation by the running time of the modified implementation.

As the experimental results show, the proposed heuristic decreases the size of the search space to great extent compared to random order of target instance selection, up to 94%, and introduces considerable speedups, up to 9.

Two extreme results are with the *Sister* and *Wife* data sets. In order to understand why those data sets behave differently than the others, we analyzed the coverage of the induced concept descriptors. In the *Sister* data set, there are

Table 2. Experimental results

Data Set	Ascending Order		Random Order		Improvement	
	# Queries	Duration	# Queries.	Duration	Queries	Speedup
Aunt	59925	110	137592	227	56%	2.06
Brother	294080	124	331602	149	11%	1.20
Daughter	117834	130	162226	189	27%	1.45
Father	5273	4	38425	36	86%	9.00
Husband	4064	3	29853	23	86%	7.66
Mother	4534	5	22672	29	80%	5.80
Nephew	55705	66	160878	218	65%	3.30
Niece	99211	175	135385	205	27%	1.17
Sister	578977	374	613422	546	6%	1.45
Son	6701	9	52821	34	87%	3.77
Uncle	40530	59	136305	236	70%	4.00
Wife	4098	4	67810	31	94%	7.75
Same Generation	104401	153858	65	99	47%	1.7
Mesh	470424	1376588	204	702	66%	3.4
PTE	594394	648318	900	1327	8%	1.47

**Fig. 1.** Behavior of the proposed heuristic for different data sets

47 target instances, and C^2D finds concept descriptors that define 31 of them at the end of the first iteration. The system, then, performs 16 more iterations to find concept descriptors to explain those remaining 16 target instances but fails to find some. On the other hand, for the *Wife* data set, the system finds solution clauses that explain all the target instances at the end of the first iteration.

In Figure 1, we graphically demonstrate the number of the queries executed to evaluate the search space for those two data sets and the *Aunt*. As the figure indicates, initial hypothesis set generated by the heuristic is always smaller than

the of the random order approach, and the resulting search space is smaller, as well. It should also need to be noted about Figure 1 that the solution clauses for the *Sister* data set is found at epoch 3, the following trails fail to find any solution clauses. As the figure indicates at epoch 3, the number of queries executed with the modified system is less than the one with random target instance selection.

When compared to random order, the proposed heuristic finds solution clauses with the same coverage values. This result also suggests that the proposed heuristic can improve efficiency of concept discovery systems without loss in the quality of induced concept descriptors.

We conducted some other experiments to analyze how the performance of the proposed heuristic would be affected when several target instances have the same global frequency and they are inter-sorted based on the frequencies of their arguments. The experimental results showed that different orderings have neglectable affect on the overall performance. As an example 581525 queries (the same value, 581525 queries in Table 2) were executed for the *Sister* data set when such tie cases were sorted in ascending order based on their least frequently appearing argument, and 4097 queries were executed (in comparison to 4098 queries in Table 2) for the *Wife* data set when such target instances were sorted in descending order based on their most frequent argument.

6 Conclusion

In this work we proposed a heuristic to select a target instance that is most likely to form a smaller initial hypothesis set and a smaller search space. The heuristic is based on counting the frequencies of the target instances in background. Each fact in the target instance is assigned with a global frequency value which is calculated by summing up frequencies of its arguments. The heuristic proposed selects the fact with the smallest frequency value as the target instance to build the initial search space. Experimental results showed that the proposed heuristic works well and provides reduction on the search space up to 94% and speedup up to 9 when compared to the runs that randomly selects a target instance.

As a future work we plan to extend the experiments with data sets that belong to different learning problems.

References

1. Muggleton, S.: Inductive Logic Programming. In: The MIT Encyclopedia of the Cognitive Sciences (MITECS). MIT Press (1999)
2. Dzeroski, S.: Multi-relational data mining: An introduction. *SIGKDD Explorations* 5(1), 1–16 (2003)
3. Muggleton, S., Feng, C.: Efficient induction of logic programs. In: *Proceedings of the 1st Conference on Algorithmic Learning Theory*, pp. 368–381. Springer/Ohmsma (1990)
4. Zelezny, F., Srinivasan, A., David Page Jr., C.: Randomised restarted search in ilp. *Machine Learning* 64(1-3), 183–208 (2006)

5. Serrurier, M., Prade, H.: Improving inductive logic programming by using simulated annealing. *Information Sciences* 178(6), 1423–1441 (2008)
6. Kavurucu, Y., Senkul, P., Toroslu, I.H.: Ilp-based concept discovery in multi-relational data mining. *Expert Syst. Appl.* 36(9), 11418–11428 (2009)
7. Nassif, H., Page, D., Ayvaci, M., Shavlik, J., Burnside, E.S.: Uncovering age-specific invasive and dcis breast cancer rules using inductive logic programming. In: *Proceedings of the 1st ACM International Health Informatics Symposium*, pp. 76–82. ACM (2010)
8. Nassif, H., Al-Ali, H., Khuri, S., Keirouz, W., Page, D.: An inductive logic programming approach to validate hexose binding biochemical knowledge. In: De Raedt, L. (ed.) *ILP 2009*. LNCS, vol. 5989, pp. 149–165. Springer, Heidelberg (2010)
9. Amini, A., Shrimpton, P.J., Muggleton, S.H., Sternberg, M.J.: A general approach for developing system-specific functions to score protein–ligand docked complexes using support vector inductive logic programming. *Proteins: Structure, Function, and Bioinformatics* 69(4), 823–831 (2007)
10. Fonseca, N.A., Pereira, M., Santos Costa, V., Camacho, R.: Interactive discriminative mining of chemical fragments. In: Frasconi, P., Lisi, F.A. (eds.) *ILP 2010*. LNCS, vol. 6489, pp. 59–66. Springer, Heidelberg (2011)
11. Konstantopoulos, S.: A data-parallel version of Aleph. *CoRR abs/0708.1527* (2007)
12. Mutlu, A., Senkul, P.: Improving hit ratio of ilp-based concept discovery system with memoization. *The Computer Journal* (2012), doi:10.1093/comjnl/bxs163
13. Blockeel, H., Dehaspe, L., Demoen, B., Janssens, G., Vandecasteele, H.: Improving the efficiency of inductive logic programming through the use of query packs. *Journal of Artificial Intelligence Research* 16, 135–166 (2002)
14. Tausend, B.: Representing biases for inductive logic programming. In: *Proceedings of the 7th European Conference on Machine Learning*, Catania, Italy, April 6–8, pp. 427–430 (1994)
15. Kavurucu, Y., Senkul, P., Toroslu, I.H.: Concept discovery on relational databases: New techniques for search space pruning and rule quality improvement. *Knowl.-Based Syst.* 23(8), 743–756 (2010)
16. Srinivasan, A.: *The Aleph Manual* (1999), <http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/> (accessed April 06, 2013)
17. <http://www.cs.utexas.edu/ftp/mooney/forte/> (accessed April 10, 2013)
18. Dolšák, B., Bratko, I., Jezernik, A.: Finite element mesh design: An engineering domain for ILP application. In: *Proceedings of the 4th International Workshop on Inductive Logic Programming*, Bonn, Germany, Gesellschaft für Mathematik und Datenverarbeitung MBH, September 12–14, pp. 305–320 (1994)
19. Srinivasan, A., King, R.D., Muggleton, S.H., Sternberg, M.: The predictive toxicology evaluation challenge. In: *IJCAI 1997: Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pp. 1–6 (1997)

Extracting Sequential Patterns Based on User Defined Criteria

Oznur Kirmemis Alkan and Pinar Karagoz

Computer Engineering Department
Middle East Technical University (METU), 06800, Ankara, Turkey
{oznur.kirmemis, karagoz}@ceng.metu.edu.tr

Abstract. Sequential pattern extraction is essential in many applications like bioinformatics and consumer behavior analysis. Various frequent sequential pattern mining algorithms have been developed that mine the set of frequent subsequences satisfying a minimum support constraint in a transaction database. In this paper, a hybrid framework to sequential pattern mining problem is proposed which combines clustering together with a novel pattern extraction algorithm that is based on an evaluation function, which utilizes user-defined criteria to select patterns. The proposed solution is applied on Web log data and Web domain, however, it can work in any other domain that involves sequential data as well. Through experimental evaluation on two different datasets, we show that the proposed framework can achieve valuable results for extracting patterns under user defined selection criteria.

Keywords: equential Pattern, User-defined selection criteria, Clustering, PatternFindBF, Web Usage Pattern.

1 Introduction

There is a huge growth of data sources on the Internet every day which can come either from the Web content, represented by the Web pages publicly available, or from the Web usage, represented by the log information daily collected by all the servers around the world. Web Usage Mining, which is the topic of concern, uses the knowledge existing in server log files that are collected when users access Web servers [28].

Web Usage mining approaches include Clustering, Association Rule Mining, Classification and Sequential Pattern Mining where the extracted patterns are generally used for recommendation and next page prediction purposes [25]. The sequential property of Web log data becomes highly valuable when extracting usage patterns is considered. Sequential pattern mining, which is the main concern of the study presented, has been researched extensively since first introduced by Agrawal [30]. In this area, Apriori-based algorithms are studied before Pattern-growth algorithms was developed [5]. Apriori-based techniques [7, 30] lacks efficiency and effectiveness in the sense that they generate huge candidate sequences. In addition, the dataset should be scanned repeatedly in order to check a large set of candidates by

some method of pattern matching. To overcome these problems, pattern-growth techniques [16, 24, 29] have been developed. Sequential pattern growth algorithms mine the complete set of frequent patterns using a divide and conquer technique in order to reduce the search space without generating all the candidates.

These algorithms treat every frequent traversal sequence equally and use only the support constraint to prune the combinatorial search space of different sequential patterns. In other words, there is no value assigned to extracted patterns, which will make one more valuable or important than another except their frequency values. However, when real world datasets are considered, support may not reflect the real value of a sequence from the perspective of the pattern user. For instance, considering Web log data, a shopping Web site owner may be interested in the shopping sequences that exist in many sessions of distinct users. Assume that web site's log data has 1000 sessions of shopping sequences, 200 of them belong to a single customer who loves shopping and buys similar items. In that case, depending on the support, frequent patterns may only exist in that customer's transactions, which will not be interesting for the web site owner. In other words, a mechanism is necessary such that all of the frequent sessions are not be treated equally and support value is not the only parameter to prune the search space.

In this work, we propose a novel sequential pattern extraction algorithm, PatternFindBF, that utilizes a user defined evaluation function and analyze its performance. In [6], the algorithm is analyzed for its performance for recommendation. In this work, the emphasis is on user-defined selection criteria and we evaluate the algorithm for extracting patterns fulfilling the given criteria. As the application domain, we use Web log data in the current version of the solution; however, the solution can easily be applied to any other sequential data. Our framework uses an evaluation function that is defined on the basis of the needs of the pattern users. For the current system, the sequences of visiting pages, the time spent on the pages, and the distinct number of users accessing the pages are used by the evaluation function. The solution proposed is a hybrid framework in the sense that it combines clustering with pattern extraction algorithm. First, different navigation behaviors are grouped through clustering. After related session groups are identified, the system utilizes PatternFindBF algorithm to construct patterns. Since the current version of the solution is evaluated using the Web log data, the extracted patterns are the usage patterns that reveal information about how users traverse the Web site.

We present a detailed experimental evaluation of our solution where comparison with a well-known sequential pattern mining algorithm, PrefixSpan [24] is presented. The experimental results show that this novel framework gets very satisfactory results and using distinct number of users in the evaluation process improves the accuracy in addition to the frequency of co-occurrence of Web pages in the user sessions. Clustering the sessions and using a limit on the number of child nodes produced at each step of PatternFindBF enables us to control and reduce the search space and the running time for extracting usage patterns.

The rest of the paper is organized as follows: In Section 2, summary of the related work is presented. In Section 3, the proposed framework is described in detail.

Next, in Section 4, the evaluation of the system and the discussion of the results are given. Finally, concluding remarks are presented in Section 5.

2 Related Work

Learning user navigation behaviors for Web page recommendation has attracted much attention in the community of data mining. Markov model and its variants have been proposed to model user behavior in many existing studies [4, 23] in addition to Association Rule Mining [1], pattern extraction under constraint [3], symbolic data analysis tools [17], semantic knowledge based solutions[2] and graph-based techniques [10]. Web surfing has become an important activity for many consumers and a Web recommender that models user behavior by constructing a knowledge base using temporal Web access patterns is proposed in [31]. As in the case for Web page recommendation, various attempts have been exploited to achieve Web page access prediction by preprocessing Web server log files and analyzing Web users navigational patterns [26].

In [8], GSP algorithm is described which extracts sequential patterns based on an Apriori like approach by generating all candidate sequences. However, this approach is inefficient and ineffective. To overcome this problem, the database projection growth based approach, FreeSpan [29], was developed. FreeSpan outperforms GSP algorithm, but FreeSpan may generate any substring combination in a sequence. The projection in FreeSpan must keep all sequences in the original sequence database without length reduction. PrefixSpan [24] increases efficiency in pattern growth by examining only the prefix subsequences and project only their corresponding suffix subsequences into projected databases. Different from these techniques, our solution uses an evaluation function in order to select the best patterns. Therefore, any data available, for instance, some sort of explicit knowledge provided from the users, can easily be integrated into the evaluation function, so that the patterns that are considered to be more valuable can be selected.

As mentioned, we have used clustering so as to group sessions into related units. Web data clustering has also been studied widely before and we can broadly categorize solutions in this area into (i) user sessions-based [9, 11, 31] and (ii) link-based [12, 13, 14]. The former aims to group user navigation sessions having similar characteristics. The latter, however, treats the Web site as a directed graph and the goal is to cluster the Web pages with similar content. Xie and Phoha [15] are the first to suggest that the focus of Web Usage Mining should be shifted from single user sessions to group of user sessions and they apply clustering for identifying such partitions of similar sessions. In [21], a technique for finely tuning users clusters' based on similar web access patterns on their usage profiles by approximating through least square approach is presented. Nasraoui et al. in [18], couples Fuzzy Artificial Immune System and clustering techniques to improve the user profiles obtained through clustering. The work described in [19], combines association rule mining and clustering into a method called association rule hypergraph partitioning. In [20], the authors propose a recommendation model where they take the visiting order of the current user into account and the active user session is assigned to the most similar cluster of user sessions according to a similarity measure.

3 Sequence Pattern Extraction under User Defined Criteria

The proposed pattern extraction algorithm takes a collection of sessions as input and generates extracted usage patterns. In this paper, we assume that sessions are already constructed from the raw web log data by using conventional session construction techniques [22, 27]. A hybrid approach is employed for the pattern discovery phase that combines clustering with a novel pattern extraction solution.

Clustering. Clustering is used for partitioning the set of sessions based on the user interests in terms of the Web page requests. For the clustering process, a session-pageview matrix is constructed where each column is a pageview and each row is a session of a user represented as a vector. The matrix values are 0 if in that session the corresponding web page is not visited, or an integer greater than zero, which represents the number of times that the Web page is visited in that session. Clustering process takes this session-pageview matrix and runs K-Means algorithm in order to find groups of sessions that exhibit similar navigation behavior.

Pattern Extraction. For each constructed cluster, sequence patterns are extracted by using a novel search oriented approach that involves a user-defined evaluation function. Different from many of the existing Web usage mining techniques, usage patterns are discovered in a stepwise manner by considering the evaluation criteria that selects to include the best representative accesses of the user interests in the discovered patterns.

The algorithm, as displayed in Figure 1, constructs patterns in a recursive manner. At each recursive call, one level of the search tree is built, which corresponds to extending the length of the patterns with one more item. At each level of the tree, at most BF (branching factor) number of nodes exists. These nodes keep the best BF patterns in terms of the evaluation function. The output is the patterns constructed for each distinct cluster.

Evaluation Function. The evaluation function examines a pattern and it assigns a value to it. The evaluation function is the part of the algorithm where user-defined selection criteria is incorporated into pattern generation. It may be any function that gets a sequence and returns a value for it with respect to the given collection of data. In this paper, we used an evaluation function where the value of a pattern depends on the number of sessions in the corresponding cluster that the pattern exists (frequency within the cluster) and the number of distinct users that traverse the pattern in their sessions, and the total duration of the pattern in user sessions (calculated as the sum of the visiting page time of all the accesses from the start access of the path to the final access). This evaluation function calculates the normalized values for each of these three dimensions and the overall value of a pattern is weighted average of them. The weights of the dimensions are taken as three parameters: $w_{session_count}$, w_{user_count} and $w_{duration}$.

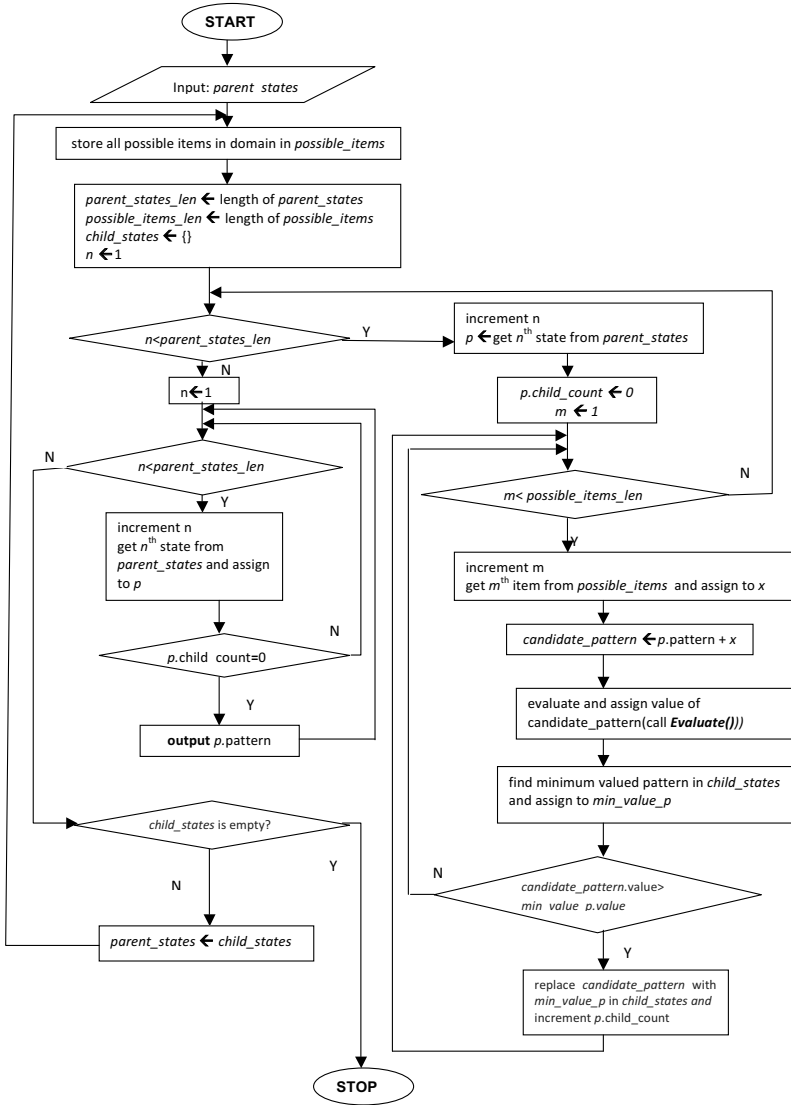


Fig. 1. PatternFindBF algorithm

The rationale behind this pattern selection criteria can be explained as follows: Web site owner wants to emphasize number of distinct users a frequent pattern is used by and the time spent along the path as well as the frequency of it. Time spent on the pattern is an indicator that the user is interested in the page and stays for a certain

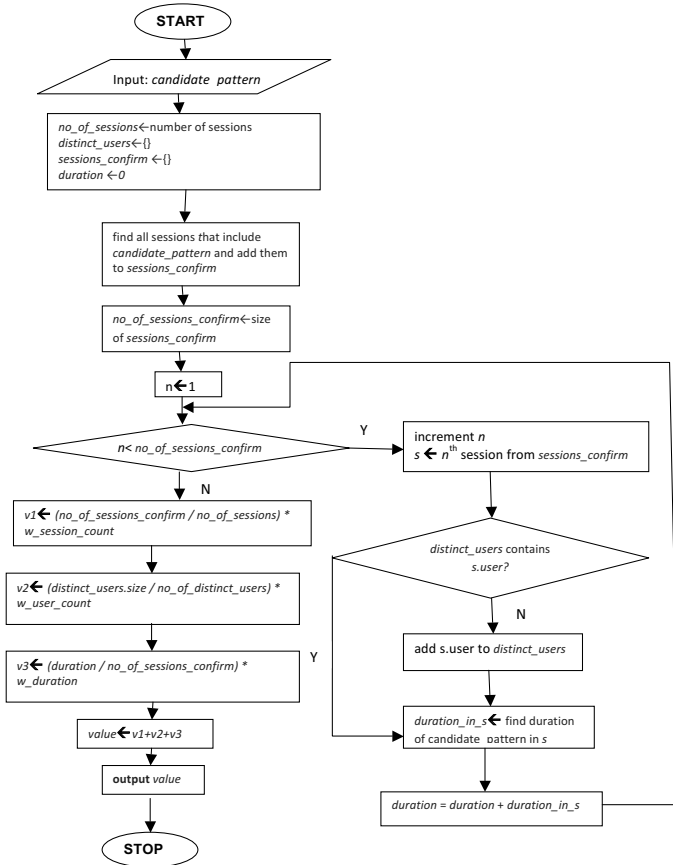


Fig. 2. Evaluate(): Sample evaluation function

amount of time to examine the content. The flowchart of the evaluation function is given in Figure 2.

Assume that we have a usage pattern (A, B, C) to evaluate, and 3 user sessions where this path exists {user1,(D-F-A-H-J-H-I-B-F-C-P)}, {user2,(A-B-F-U-C)}, and {user1,(T-A-F-H-B-C-A-D)}. Assume further that, the visiting page time for each access in all the sessions is 1, to keep the example simple. Then the number of sessions that this path exists is 3. The number of distinct users that covers this path in their access sequences is 2 and total duration of this path in sessions is 8(A-H-J-H-I-B-F-C) + 5(A-B-F-U-C) + 5(A-F-H-B-C), which makes 18. We then calculate the value of this path using the formula given in Figure 2. It is important to note that such an evaluation function can easily be adapted to different criteria and different domains other than Web.

4 Experimental Evaluation

In this section, the evaluation process of the system is detailed by giving information about the datasets used, the methodology and the metrics conducted, and the results collected throughout the experiments.

4.1 Data Sets

In this research, we use NASA and CENG datasets for the evaluation phase. The NASA dataset (<http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html>) is from the NASA Kennedy Space Center server over the months of July and August 1995. It originally contains 3,461,612 requests. After the data preparation phase, we have 737,148 accesses, 90,707 sessions, 2459 distinct requests, and 6406 distinct users. The CENG dataset (www.ceng.metu.edu.tr) is from the Computer Engineering (CENG) Department of Middle East Technical University (METU). The dataset consists of many sub-Web sites including Web pages of individuals (i.e., students, teachers), newsgroups, and courses. It contains web server logs from 03.07.2011 to 13.11.2011. The dataset originally contains 7,041,032 accesses. After preprocessing, there remains 1,752,771 page views and 304,567 distinct sessions.

4.2 Experimental Results

In this experiment, we aim to compare the values of the patterns identified by our framework and by frequent sequential pattern mining in terms of the value calculation used in our solution. In other words, we compare the patterns extracted by PrefixSpan and our framework using the calculation done by the evaluation function in the PatternFindBF so as to select higher valued patterns. Figure 3 (a) shows the values of two groups of identified patterns, one by PrefixSpan and the other by our solution for the NASA dataset. The x axis refers to the top n number of frequent versus high-valued patterns selected from the two groups, while the y axis shows the sum of the values of the top n patterns. In addition, Figure 3 (b) shows the average values of patterns with different lengths from PrefixSpan and our framework for the NASA dataset where the x axis refers to the lengths of patterns and the y axis shows the average values per pattern. When both of the results are considered, it can be concluded that our solution can identify higher valued patterns, and it can extract top patterns with higher average values per pattern than PrefixSpan.

Similar results are obtained for CENG dataset as given in Figure 4 in the sense that the proposed algorithm can generate better patterns in terms of target criteria. However, the behavior for average value per pattern is different in each dataset. As seen in Figure 4 (a), for the proposed method the amount of target value accumulated is always higher than that of PrefixSpan and increases steadily.

Figure 4 (b) shows the change in average target value as the length of the generated patterns increases. This time the behavior is not steady and on the overall there is an increase. However, since the evaluation function involves three dimensions, it is not easy to tell directly which one is more effective on the overall value.

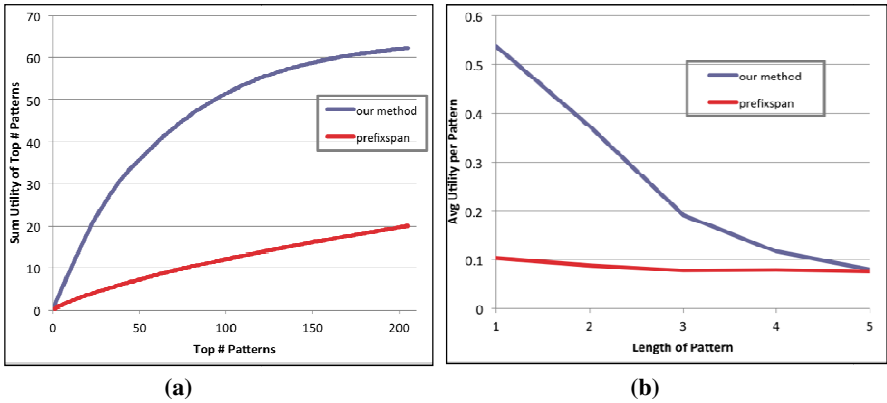


Fig. 3. Comparison of value results of our solution and PrefixSpan for the NASA dataset

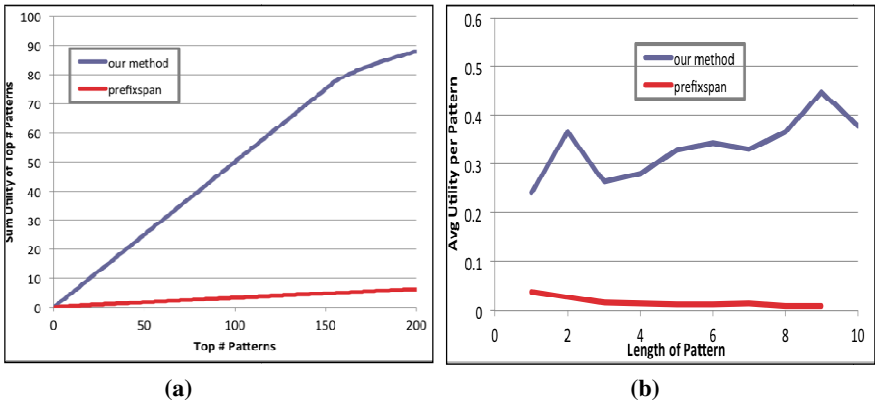


Fig. 4. Comparison of value results of our solution and PrefixSpan for the CENG dataset

5 Conclusion

In this paper, we have presented a novel framework for sequential pattern mining problem. The framework is implemented and evaluated using the Web log data, however, it can work in any other domain as well. The solution is based on combining clustering with pattern extraction algorithm. The role of clustering in the framework is to group user sessions according to similar navigational behaviors that increase the accuracy of the system as shown by the experiments.

For the pattern extraction phase, the solution utilizes user defined criteria to select patterns, instead of using a support limit, as it is the case for existing sequential pattern mining algorithms. The system is evaluated using two different datasets in term of accumulating scores for user-defined criteria. The experimental results show that, the proposed solution can outperform PrefixSpan algorithm in terms of the evaluation

criteria used in the system. Our future work is on evaluating the solution using sequential datasets from different domains under various user-defined criteria.

Acknowledgements. This work is supported by METU BAP project BAP-03-12-2013-001.

References

1. Leung, C.W., Chan, S.C., Chung, F.: A Collaborative Filtering Framework Based on Fuzzy Association Rules and Multiple-Level Similarity. *Knowl. Inf. Syst.* 10(3), 357–381 (2006)
2. Senkul, P., Salin, S.: Improving Pattern Quality in Web Usage Mining by Using Semantic Information. *Knowl. Inf. Syst.* 30(3), 527–541 (2011)
3. Shyu, M., Haruechaiyasak, C., Chen, S.: Mining User Access Patterns with Traversal Constraint for Predicting Web Page Requests. *Knowl. Inf. Syst.* 10(4), 515–528 (2006)
4. Bonnin, G., Brun, A., Boyer, A.: A Low-Order Markov Model Integrating Long-Distance Histories for Collaborative Recommender Systems. In: *Proceedings of the 14th International Conference on Intelligent User Interfaces*, Florida, USA, pp. 57–66 (2009)
5. Mooney, C.H., Roddick, J.F.: Sequential Pattern Mining – Approaches and Algorithms. *ACM Computing Surveys (CSUR) Surveys Homepage Archive* 45(2) (2013)
6. Kirmemis Alkan, O., Karagoz, P.: Assisting Web Site Navigation Through Web Usage Patterns. In: *IEA/AIE 2013*, Amsterdam, Netherlands (June 2013)
7. Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) *EDBT 1996*. LNCS, vol. 1057, pp. 1–17. Springer, Heidelberg (1996)
8. Ren, J.D., Cheng, Y.B., Yang, L.L.: An Algorithm for Mining Generalized Sequential Patterns. In: *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 1288–1292 (2004)
9. Banerjee, A., Ghosh, J.: Clickstream Clustering Using Weighted Longest Common Subsequences. In: *Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining* (2001)
10. Daoud, M., Lechani, L., Boughanem, M.: Towards a Graph-Based User Profile Modeling for a Session Based Personalized Search. *Knowl. Inf. Syst.* 21(3), 365–398 (2009)
11. Kothari, R., Mittal, P.A., Jain, V., Mohania, M.K.: On Using Page Cooccurrences for Computing Clickstream Similarity. In: *Proceedings of the 3rd SIAM International Conference on Data Mining*, San Francisco, USA (2003)
12. Eiron, N., McCurley, K.S.: Untangling Compound Documents on the Web. In: *Proceedings of ACM Hypertext*, pp. 85–94 (2003)
13. Flake, G.W., Lawrence, S., Giles, C.L., Coetzee, F.: Self-Organization and Identification of Web Communities. *IEEE Computer* 35(3) (2002)
14. Greco, G., Greco, S., Zumpano, E.: Web Communities: Models and Algorithms. *World Wide Web* 7(1), 58–82 (2004)
15. Xie, Y., Phoha, V.V.: Web User Clustering from Access Log Using Belief Function. In: *Proceedings of the First International Conference on Knowledge Capture*, pp. 202–208. ACM Press (2001)
16. Pinto, H., Han, J., Pei, J., Wang, K.: Multi-dimensional Sequence Pattern Mining. In: *CIKM* (2001)

17. Bezerra, B.L.D., Carvalho, F.A.T.: Symbolic Data Analysis Tools for Recommendation Systems. *Knowl. Inf. Syst.* 21(3), 385–418 (2010)
18. Nasraoui, O., Gonzalez, F., Dasgupta, D.: The Fuzzy Artificial Immune System: Motivations, Basic Concepts, and Application to Clustering and Web Profiling. In: *Proceedings of the World Congress on Computational Intelligence (WCCI) and IEEE International Conference on Fuzzy Systems*, pp. 711–716 (2002)
19. Mobasher, B., Dai, H., Tao, M.: Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. *Data Mining and Knowledge Discovery* 6, 61–82 (2002)
20. Gunduz, S., Ozsu, M.T.: A Web Page Prediction Model Based on Clickstream Tree Representation of User Behavior. In: *SIGKDD 2003, USA*, pp. 535–540 (2003)
21. Patil, S.S.: A Least Square Approach to Analyze Usage Data for Effective Web Personalization. In: *Proceedings of International Conference on Advances in Computer Science*, pp. 110–114 (2011)
22. Cooley, R., Mobasher, B., Srivastava, J.: Web Mining: Information and Pattern Discovery on the World Wide Web. In: *International Conference on Tools with Artificial Intelligence*, Newport Beach, pp. 558–567. IEEE (1997)
23. Yang, Q., Fan, J., Wang, J., Zhou, L.: Personalizing Web Page Recommendation via Collaborative Filtering and Topic-Aware Markov Model. In: *Proceedings of the 10th International Conference on Data Mining, ICDM, Sydney*, pp. 1145–1150 (2010)
24. Pei, J., Han, J., Mortazavi-Asi, B., Pino, H.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In: *ICDE 2001* (2001)
25. Mobasher, B.: Data Mining for Web Personalization. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *Adaptive Web 2007*. LNCS, vol. 4321, pp. 90–135. Springer, Heidelberg (2007)
26. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.: Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGDD Explorations* 1(2), 12–23 (2000)
27. Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. *Knowl. Inf. Syst.* 1(1), 12–23 (1999)
28. Etzioni, O.: The World Wide Web: Quagmire or gold mine? *Communications of the ACM* 39(11), 65–68 (1996)
29. Han, J., Pei, J., Mortazavi-Asi, B., Chen, Q., Dayal, U., Hsu, M.C.: FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining. In: *SIGKDD 2000* (2000)
30. Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: *ICDE* (1995)
31. Fu, Y., Sandhu, K., Shih, M.Y.: Clustering of Web Users Based on Access Patterns. In: *Proceedings of WEBKDD* (1999)

Sequence Alignment Adaptation for Process Diagnostics and Delta Analysis

Eren Esgin¹ and Pınar Karagoz²

¹ Middle East Technical University, Informatics Institute,

² Middle East Technical University, Computer Engineering Department,
065531 Ankara, Turkey
eesgin@ii.metu.edu.tr, karagoz@ceng.metu.edu.tr

Abstract. Business process management (BPM) paradigm gains growing attention by providing generic process design and execution capabilities. During execution, many business processes leave casual footprints (event logs) at these transactional information systems. *Process mining* aims to extract business processes by distilling event logs for knowledge. *Sequence alignment* is a technique that is frequently used in domains including bioinformatics, language/text processing and finance. It aims to arrange structures, such as protein sequences to identify similar regions. In this study, we focus on a hybrid quantitative approach for performing *process diagnostics*, i.e. comparing the similarity among process models based on the established *dominant behavior* concept and *Needleman-Wunsch algorithm*.

Keywords: Process Mining, Sequence Alignment, Process Diagnostics, Dominant Behavior, Needleman-Wunsch Algorithm.

1 Introduction

While contemporary information systems are intensively utilized in the enterprises, their leverage effect in automating business processes is limited by difficulties faced at process design phase [2]. Actually crucial problems (e.g. deadlocks, hidden and repeated activities) are resulting from the discrepancies between process design and process enactment [4, 5].

Process design is influenced by personal perceptions, e.g. reference process models are often normative in the sense that reflect what *should* be done rather than describing the actual case [3]. As a result, proposed reference models tend to be rather incomplete, subjective and at a too coarse-grained level. In [6], authors argue that IT research is subject to subjectivistic–objectivistic dilemma. For instance, introducing an *enterprise resource planning* (ERP) to transform existing business process corresponds to an objectivistic view since it assumes the technology to be the sole relevant change factor. On the other hand, eliciting requirements for the ERP implementation and building the system upon this baseline is the subjectivistic perspective [6].

Process mining is proposed as the remedy to handle these discrepancies by distilling significant patterns from the *event logs* and discovering the business process

model automatically [2, 7, 8, 9, 10, 11]. Unlike to traditional *design-centric* approach, process mining is not biased and restrictive by normative perceptions [3], because event logs reflect what process participants are doing at the operational level, not just what management is suggesting.

When a process design from an enterprise is provisioned to process participants, process participants refine it on their demands. According to Gidden's theory, the new organization emerges through process participants' actions in their new environment and it is shaped by the resulting set of work activities realized to achieve the business process goals [16]. At that point, important information to improve an existing process design is *where* process participants deviate from the intended process definition.

In BPM, this activity is known as *process diagnostics*, i.e. encompassing process performance analysis, anomaly detection, diagnosis, inspection of interesting patterns [17]. In several other domains sequence alignment is employed for comparing sequences. As one of such domains, bioinformatics aims at increasing the understanding of biological processes and entails the application of sequence analysis to predict the biological function of a gene, find the evolution distance and common regions (i.e. repeats) in homologous genomes [22].

In this work, we propose a hybrid quantitative approach that exploits sequence alignment for *delta analysis*, i.e. comparing the actual process, represented by a process model constructed through process mining, with *prescriptive* reference process model [11]. The approach initially derives *consensus activity sequence* that captures the major *dominant behavior*. At this step, the method in [14] is used for constituting the backbone sequence for the underlying business process. As the second phase, optimal alignment among derived consensus activity sequences is built up using *Needleman-Wunsch algorithm*, which is fundamentally a dynamic programming (DP) algorithm for finding the optimal alignment between two amino-acid sequence with the maximum alignment score [22]. The main contribution of this paper lies in the second phase.

This paper is organized as follows: Section 2 includes literature review. Section 3 highlights the major aspects and prior studies on process discovery. Section 4 introduces the design of the proposed quantitative approach in measuring the similarity between the process models due to process diagnostics paradigm. Section 5 emphasizes experiments based on intuitive similarity judgment and finally Section 6 summaries the concluding remarks.

2 Related Work

The equivalence of process models are usually subject to *verification* such as trace equivalence. This stream is based on a comparison of the sets of completed event logs likewise in [17]. Kleiner does not wish to discover a graphical process model but use such logs to check for deviation from a prescribed business process model, since he assumes that the activities of a mined process model are usually on a low level of abstraction.

Esgin and Senkul [23] propose a distance metric, which is built on the *vector model* from information retrieval and an abstraction of process behavior as process triple. This metric takes into structural and behavioral perspectives into account.

Cook and Wolf [9] present an approach for delta analysis in the context of software processes. They use AI algorithms to compare process models and compute the level of correspondence. Additionally they assume that there exists any difference at abstraction level of event logs and discovered process model.

Actually, *behavioral semantics* can lead to performance problems due to large sets of traces and state explosion. In [19], an approximation on behavioral semantics is given by the concept of causal footprint. In this study, instead of computing the similarity between each pair of process models, these models are represented as a point at Euclidian distance space. Hence the underlying problem is reduced to the nearest-neighbor problem.

A second way of defining behavioral semantic is monitoring the states in which the process model can be. This idea is realized by Nejati et al. [20] by taking the state-space of two process models and check if they can simulate one another. By counting such states, we can measure how dissimilar two process models are.

Another perspective in delta analysis is the *graph theory*, which is a useful means to analyze the process definitions. Especially, Bunke in [18] has shown that with generic graphs, the maximum common sub-graph (MCS) is equivalent to computing the graph edit-distance emphasized in [21]. This MCS is the baseline to measure the common activities and transitions of workflow processes in [1].

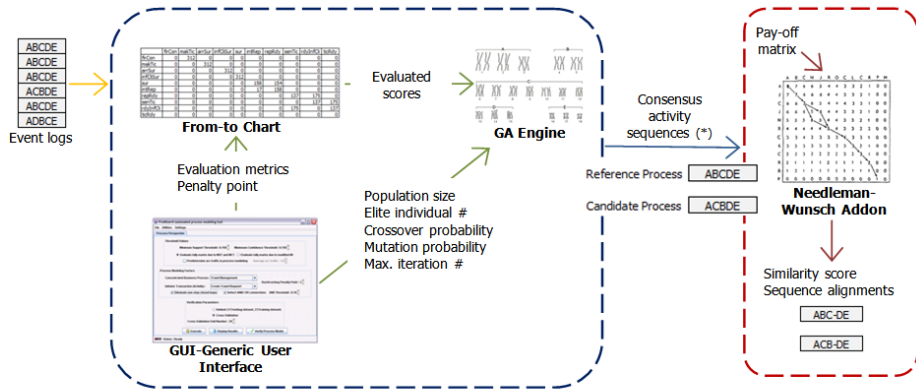
3 Major Aspects

A new approach for process discovery based on using from-to chart is introduced in [12]. In this approach, *from-to chart*, which is a square symmetrical matrix used in monitoring material handling routes on the production floor [13], is used for analyzing the event logs. The underlying approach inherits this tool from facility layout problem (FLP) domain and adapts it in a distinct field, i.e. *process discovery*, as the basic bookkeeping material in monitoring transitions among activities occurred in process instances and concluding if there exists any specific order of the occurrences for representing in process model [12].

This approach is further improved in [14] by several enhancements. In the prior version, chart rearrangement was performed in a permutative manner, which led to an exponential increase in processing time [12]. Hence in [14], the runtime complexity of the approach is improved by adopting *Genetic Algorithms* (GA), which are adaptive methods used to solve search and optimization problems [15].

Unlike prior studies [12, 14, 23], the main idea of this work is to compare process models relative to *consensus activity sequences* containing *dominant behavior* (i.e. common subsequence of activities in event log that are found to recur within a process instance or across process instances with some domain significance). Thus the adaptation of *sequence alignment* in bioinformatics to process mining has created an altogether new perspective to delta analysis; deviations and violations are uncovered by analyzing just consensus activity sequences (thereby avoiding the requirement for well-structured process models).

Unfortunately, most equivalence notions concentrate on an atomic *true-false* answer. In reality there will seldom be a perfect fit. Hence we are interested in the



(*) minimum two sequences such that, one sequence for base business process, one for candidate process.

Fig. 1. Overview for the Proposed Approach

degree of similarity. In order to do so, Needleman-Wunsch algorithm is applied to quantify the similarities and deviations. The overview of proposed approach is given in Figure 1.

4 Proposed Approach

4.1 Constructing from-to Chart from Event Logs

The starting point of the proposed approach is the creation of a so-called FROMTOCHART table by retrieving the activity types from the *event logs* and populating FROMTOCHART table. For populating the table, event logs are arranged by process instances (e.g. caseID) and then ordered by timestamp in ascending order. Then, predecessor and successor are parsed for each transition in activity streams and the current score at (predecessor, successor)th element in FROMTOCHART table is incremented by one.

4.2 Evaluating the Scores at from-to Chart

In traditional from-to chart implementation, total score of each element is directly taken into consideration in rearrangement of the matrix. However, by pruning down the *weak* scores prior to rearrangement, it is aimed to eliminate their effect on the fittest activity sequence in the proposed approach [12].

Basically there are three evaluation metrics emphasized in [12]: *confidence for from-to chart* (confidence FTC), *support for from-to chart* (support FTC) and *modified lift*. These metrics act as the major stick yard to control the level of robustness and complexity of the discovered process model from large amounts of data. While formulating these metrics, the original formations for evaluations in association rule mining are taken as the basis [12].

4.3 Rearranging from-to Chart

This operation is the *engine component* of proposed approach, which aims to find out the *consensus activity sequence* with the minimum total moment value at FROMTOCHART table [14]. The mapping of basic GA notations into the business process modeling domain is such that; a *chromosome* possessed by an individual is represented as an *activity sequence* and each *gene* position in this chromosome corresponds to a unique *activity type*. The coarse-grained GA stages are implemented as follows:

i. Initialization. Initial population can be generated with or without a *schema*. Holland’s *schema theorem* explains the power of the GA in terms of how schemata are processed [14].

In the case of non-schema application, a Permutation class randomly generates the candidate chromosomes, which are encoded according to activity type domain alphabet. On the other hand if the initial population is generated according to the schema, the schema has to be constructed firstly. In order to construct the schema, a *transition top-list*, which holds *top n scores* that are retrieved from the from-to chart, is instantiated. Afterwards, a top-down search is performed at the transition top-list in order to construct the schema. As a result, a non-intermittent schema with the maximum length of $\lfloor \text{activity type domain} \rfloor / 3$ is constructed.

ii. Fitness Score Calculation. As far as GA are concerned, it is better to have higher fitness scores to provide more opportunities especially in selection stage. Therefore the inverse of the moment function is used as the denominator of the fitness function to search for the minimum moment [14]. The numerator of the fitness function is set to the total scores that are marked at the from-to chart.

$$f(z) = \frac{\sum_{i=1}^N \sum_{j=1}^N score_{ij}}{\sum_{i \in \text{chromosomeZ}} \sum_{j \in \text{chromosomeZ}} score_{ij} \times |j - i| \times p} \tag{1}$$

Because of the moment notation, the upper-bound value for the fitness function, $f(z)$, is 1 (i.e. all transitions at from-to chart are straight-line type).

iii. Selection. As the selection method, roulette wheel selection is implemented. Roulette wheel selection is a kind of random selection type where the individual has a probability of $\text{FitnessScore}_i / \sum \text{FitnessScore}$ to be selected as a parent to mate. Since higher fitness score means higher chance to mate, the random choice is biased towards the fitness score.

iv. Crossover. Actually crossover is not always applied to all pairs of parents selected for mating such that, a default likelihood of crossover is set to $p_c = 0.80$. If the crossover is not applied, the offspring are produced by simply duplicating the parents. Otherwise, a single crossover gene position, which is in the $[1, \text{chromosomeLength}]$ interval, is randomly determined and chromosome subsets are exchanged according to this gene position.

v. Mutation. Mutation independently alters each gene value at the offspring chromosome with relative small probability (typically $p_m=0.02$). In higher order domain alphabets, in which binary coding is not appropriate, mutation and crossover framework may cause problems with chromosomes legality, e.g. multiple copies of a given activity type may occur at the offspring. Therefore we propose an alternative mutation scheme that automatically swaps the duplicate activity type with a randomly selected unobserved value. Hence a uniform chromosome that satisfies the chromosome legality is reproduced.

vi. Population Convergence. As a termination condition, if at least 95% of the individuals in the last population are in the *convergence band*, no more new population is generated. Convergence bandwidth is determined by domain expert or knowledge engineer. In order to overwhelm premature convergence, convergence ratio has to be set appropriately.

4.4 Aligning Consensus Activity Sequences

The basic idea of *Needleman-Wunsch algorithm* is to build up a global optimal alignment using previous solutions for optimal alignments of smaller consensus activity sequences. Let T_1 and T_2 be two *consensus activity sequences* generated at rearrangement step. A matrix F indexed by i and j , is constructed where the value $F(i,j)$ is the score of the best alignment between the prefix T_1^i of T_1 and the prefix T_2^j of T_2 . $F(i,j)$ is constructed recursively by initializing $F(0,0)=0$ and then proceeding to fill the matrix from top left to bottom right. It is possible to calculate $F(i,j)$ according to neighboring values, $F(i-1,j)$, $F(i-1,j-1)$ and $F(i,j-1)$. There are *three possible ways* that the best score $F(i,j)$ of an alignment up to sub-sequences T_1^i and T_2^j can be obtained:

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + S(T_1^i, T_2^j) \\ F(i-1, j) + I(T_1^i, T_1^{i-1}) \\ F(i, j-1) + I(T_2^j, T_2^{j-1}) \end{cases} \quad (2)$$

$S(T_1^i, T_2^j)$, $I(T_1^i, T_1^{i-1})$ and $I(T_2^j, T_2^{j-1})$ parameters at equation (2) stand for MATCH, MISMATCH and INDEL (i.e. insertion or deletion) multipliers determined at *pay-off matrix*. The value at the bottom right cell of the matrix, $F(|T_1|, |T_2|)$, is the similarity score for the alignment of activity sequences T_1 and T_2 . In order to find out the optimal alignment, we must *backtrack* the path of choices from (2) that led to this best score, i.e., we move from the current cell (i,j) to one of the neighboring cells from which the value $F(i,j)$ is derived.

While backtracking, a pair of symbols is added onto the front of alignment: T_1^i and T_2^j if the step is to $(i-1, j-1)$, T_1^i and the gap symbol – if the step is to $(i-1, j)$ or – and T_2^j if the step is to $(i, j-1)$. Backtracking process is terminated at the starting point $(0,0)$.

5 Experimental Results

5.1 Comparison with Intuitive Judgments

The validation of the underlying similarity measurement approach is based on the comparison of the similarity measurements of the proposed approach with the intuitive judgments of SAP consultants at various modules. Basically, the intuitive judgments are collected by a questionnaire, which consists of synthetic reference and five candidate process models given in Figure 2. SAP consultants rank the candidate process models according to the structural and behavioral similarities with reference model. Then these rankings are converted to 1-0 Likert chart.

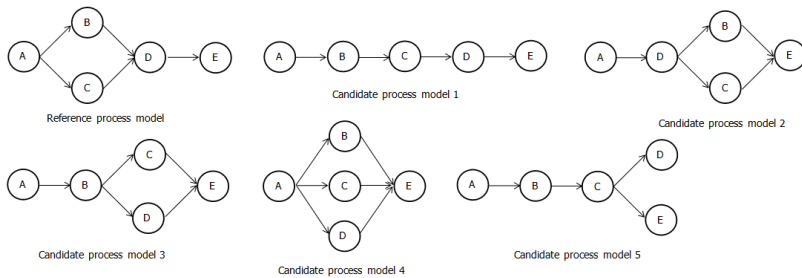


Fig. 2. Synthetic Process Models

In parallel to intuitive judgments, 300 process instances are generated for each business process by hand-simulation. In order to eliminate inductive bias, rearrangement/evaluation and alignment parameters are configured in five consecutive sets as given at Table 1 and 2.

Table 1. & 2. Rearrangement/Evaluation and Alignment Factors Configuration

	Rearrangement/Evaluation Factors					Alignment Factors (Pay-off Matrix)					
	Set 1	Set 2	Set 3	Set 4	Set 5	Set 1	Set 2	Set 3	Set 4	Set 5	
confidence FTC	0.6	0.4	0.2	0.4	0.4	1	1	1	2	5	
support FTC	0.6	0.4	0.2	0.4	0.4	MISMATCH	-1	-1	-1	-2	-5
Population Size	80	80	80	120	40	INSERT	-1	-5	0	-1	0
P(crossover)	0.8	0.8	0.8	0.8	0.8	DELETE	-1	-5	0	-1	0
P(mutation)	0.02	0.02	0.02	0.02	0.02						
reference model	DE	ABCDE	ABCDE	ABCDE	A8CDE						
candidate model 1	ABCDE	ABCDE	ABCDE	ABCDE	ABCDE						
candidate model 2	ADBCE	ADBCE	ADBCE	ADBCE	ADBCE						
candidate model 3	AB	ABCDE	ABCDE	ABCDE	A8CDE						
candidate model 4	N/A	ACBDE	ACBDE	ACBDE	ACBDE						
candidate model 5	ABC	ABCDE	ABCDE	ABCDE	ABCDE						

Gray-shaded sequences illustrate the **consensus activity sequences** (i.e. **dominant behavior**) constructed by rearrangement operation.

25 business process alignment runs are performed due to the settings at evaluation, rearrangement and alignment operations. According to reference process model alignment and similarity score at each run given at Table 3, candidate process models are ranked and then these rankings are converted to 1-0 Likert chart.

Table 3. Business Process Alignment Runs
(reference process model alignments and similarity scores)

		Rearrangement/Evaluation Factors									
		Set 1		Set 2		Set 3		Set 4		Set 5	
Alignment Factors (Pay-off Matrix)	Set 1	--DE	-1	ABCDE	5	ABCDE	5	ABCDE	5	ABCDE	5
		-D-E	-1	A-BCDE	2	A-BCDE	2	A-BCDE	2	A-BCDE	2
		DE	-2	ABCDE	5	ABCDE	5	ABCDE	5	ABCDE	5
		N/A		ABC-DE	2	ABC-DE	2	ABC-DE	2	ABC-DE	2
		DE-	-3	ABCDE	5	ABCDE	5	ABCDE	5	ABCDE	5
	Set 2	--DE	-13	ABCDE	5	ABCDE	5	ABCDE	5	ABCDE	5
		-D-E	-13	ABCDE	-1	ABCDE	-1	ABCDE	-1	ABCDE	-1
		DE	-2	ABCDE	5	ABCDE	5	ABCDE	5	ABCDE	5
		N/A		ABCDE	1	ABCDE	1	ABCDE	1	ABCDE	1
		DE-	-7	ABCDE	5	ABCDE	5	ABCDE	5	ABCDE	5
	Set 3	--DE	2	ABCDE	5	ABCDE	5	ABCDE	5	ABCDE	5
		-D-E	2	A-BCDE	4	A-BCDE	4	A-BCDE	4	A-BCDE	4
		DE--	0	ABCDE	5	ABCDE	5	ABCDE	5	ABCDE	5
		N/A		ABC-DE	4	ABC-DE	4	ABC-DE	4	ABC-DE	4
		DE---	0	ABCDE	5	ABCDE	5	ABCDE	5	ABCDE	5
	Set 4	--DE	1	ABCDE	10	ABCDE	10	ABCDE	10	ABCDE	10
		-D-E	1	A-BCDE	6	A-BCDE	6	A-BCDE	6	A-BCDE	6
		DE--	-4	ABCDE	10	ABCDE	10	ABCDE	10	ABCDE	10
		N/A		ABC-DE	6	ABC-DE	6	ABC-DE	6	ABC-DE	6
		DE---	-5	ABCDE	10	ABCDE	10	ABCDE	10	ABCDE	10
Set 5	--DE	10	ABCDE	25	ABCDE	25	ABCDE	25	ABCDE	25	
	-D-E	10	A-BCDE	20	A-BCDE	20	A-BCDE	20	A-BCDE	20	
	DE--	0	ABCDE	25	ABCDE	25	ABCDE	25	ABCDE	25	
	N/A		ABC-DE	20	ABC-DE	20	ABC-DE	20	ABC-DE	20	
	DE---	0	ABCDE	25	ABCDE	25	ABCDE	25	ABCDE	25	

Similarity measurements of the proposed approach and intuitive judgments obtained from SAP consultants are compared by dependent t-test, since hand-simulated event logs are dependent to synthetic process models, which are also the baseline for intuitive judgments. According to the t-value (0.296 versus $t_{0.05,48}$), the null hypothesis, H_0 , which states that there is no clear distinction between similarity measurement of the proposed approach and intuitive judgments obtained from SAP consultants, is accepted. This result implies that, proposed approach appropriately reflects the perceptions of knowledge engineers (i.e. *tacit process variant assumption*). On the other hand, the gap between variance figures (49.317 versus 6.017) seemingly highlights the mechanism such that; proposed approach takes more complex determinants (i.e. factors given at Tables 1 and 2) into account, while fewer (structural and tacit) features dominate intuitive judgments. The result of t-test ($\alpha=0.05$ and $CI=95\%$) is given at Table 4.

Table 4. t-test for Similarity Measurement Comparison
(Proposed Approach versus Human Judgments)

	<i>Proposed Approach</i>	<i>Intuitive Judgments</i>
Mean	3.440	3.000
Variance	49.317	6.017
Observations	25	25
Pooled Variance	27.667	
Hypothesized Mean Difference	0.000	
df	48	
t Stat	0.296	
P(T<=t) one-tailed	0.384	
t Critical one-tailed	1.677	
P(T<=t) two-tailed	0.769	
t Critical two-tailed	2.011	

5.2 Comparison with Prior Approach

Another comparison is performed with the *dissimilarity metric* given in [23]. In this work, a hybrid approach in measuring the dissimilarities between business process models is proposed such that; *dependency/frequency graph*, which is finite-state machine like block-model representation, is converted to *vector models*. The major difference of the proposed vector model from the standard vector model is the term weight assignment such that; the terms that reflect structural feature are atomic and the terms that reflect behavioral feature are represented as set [23].

Likewise in the previous analysis, the comparison of these two (dis)similarity metrics is realized by dependent t-test. According to the t-value (0.223 versus $t_{0.05,48}$), the null hypothesis, H_0 , which states that there is no clear distinction between similarity measurements with these two metrics, is accepted. Although there is a basic parallelism between proposed approach and dissimilarity metric; dissimilarity metric is totally derived from process model, while proposed approach evaluates just consensus activity sequences (avoiding the requirement for well-structured process models). The result of t-test ($\alpha=0.05$ and $CI=95\%$) is given at Table 5.

Table 5. t-test for Similarity Measurement Comparison
(Proposed Approach versus Dissimilarity Metric)

	<i>Proposed Approach</i>	<i>Prior Approach</i>
Mean	3.440	3.000
Variance	49.317	47.917
Observations	25	25
Pooled Variance	48.617	
Hypothesized Mean Difference	0.000	
df	48	
t Stat	0.223	
P(T <=t) one-tailed	0.412	
t Critical one-tailed	1.677	
P(T <=t) two-tailed	0.824	
t Critical two-tailed	2.011	

6 Conclusion

In this paper, we demonstrated that *process mining* can benefit from sequence mining techniques, which are frequently used in *bioinformatics*. Unlike prior studies [12,14,23], the main idea is to compare process models with respect to consensus activity sequences containing *dominant behavior* (i.e. typical behavior obtained on the basis of event logs). Thus the application of *sequence alignment* in bioinformatics to process mining has highlighted a new perspective to delta analysis; deviations and violations are uncovered by analyzing just consensus activity sequences (thereby avoiding the requirement for well-defined process models).

While most equivalence notions concentrate on an atomic true-false answer, we are interested in the *degree of similarity*. In order to do so, *Needleman-Wunsch algorithm* is applied to quantify the similarities and discrepancies. Actually this similarity measurement takes *structural* and *behavioral* perspectives into account. According to experimental analysis, proposed similarity metric successfully simulates the human assessment model and the results are consistent with the prior dissimilarity metric in [23].

References

- [1] Huang, K., Zhou, Z., Han, Y., Li, G., Wang, J.: An Algorithm for Calculating Process Similarity to Cluster Open-Source Process Designs. In: Jin, H., Pan, Y., Xiao, N., Sun, J. (eds.) GCC 2004. LNCS, vol. 3252, pp. 107–114. Springer, Heidelberg (2004)
- [2] van der Aalst, W.M.P., Gunther, C., Recker, J., Reichert, M.: Using Process Mining to Analyze and Improve Process Flexibility. In: Proc. of BPMDS 2006 (2006)
- [3] van der Aalst, W.M.P., Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, T.A.J.M.M.: Workflow Mining: A Survey of Issues and Approaches. *Data & Knowledge Engineering* 47(2), 237–267 (2003)
- [4] Märušter, L., Weijters, A.J.M.M.T., van der Aalst, W.M.P., van den Bosch, A.: Process Mining: Discovering Direct Successors in Process Logs. In: Lange, S., Satoh, K., Smith, C.H. (eds.) DS 2002. LNCS, vol. 2534, pp. 364–373. Springer, Heidelberg (2002)
- [5] Weijters, A., van der Aalst, W.M.P.: Process Mining Discovering Workflow Models from Event-Based Data. *Integrated Computer-Aided Engineering* (2003)
- [6] Markus, L.M., Robey, D.: Information Technology and Organizational Change: Causal Structure in Theory and Research. *Management Science* 34(5), 583–598 (1988)
- [7] Gunther, C.W., van der Aalst, W.M.P.: Process Mining in Case Handling Systems. In: Proc. of Multikonferenz Wirtschaftsinformatik 2006 (2006)
- [8] Agrawal, R., Gunopulos, D., Leymann, F.: Mining Process Models from Workflow Logs. In: Proc. of the Sixth Intern. Conf. on Extending Database Technology (1998)
- [9] Cook, J.E., Wolf, A.L.: Discovering Models of Software Processes from Event-Based Data. *ACM TOSEM* 7(3), 215–249 (1996)
- [10] Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering Workflow Models from Event-Based Data Using Little Thumb. *ICAE* 10(2), 151–162 (2003)
- [11] van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. *Transaction on Knowledge and Data Engineering* 16(9), 1128–1142 (2004)
- [12] Esgin, E., Senkul, P.: Hybrid Approach to Process Mining: Finding Immediate Successors of a Process by Using From-to Chart. In: ICMLA, pp. 664–668 (2009)
- [13] Francis, R.L., McGinnis, L.F., White, J.A.: Facility Layout and Location: An Analytical Approach. Prentice Hall, New Jersey (1992)
- [14] Esgin, E., Senkul, P., Cimenbicer, C.: A Hybrid Approach for Process Mining: Using From-to Chart Arranged by Genetic Algorithms. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010, Part I. LNCS, vol. 6076, pp. 178–186. Springer, Heidelberg (2010)
- [15] Dianati, M., Song, I., Treiber, M.: An Introduction to Genetic Algorithms and Evaluation Strategies. Univ. of Waterloo, Canada
- [16] Giddens, A.: Central Problems in Social Theory: Action Structure and Contradiction in Social Analysis. University of California Press, Berkley (1979)
- [17] van der Aalst, W.M.P.: Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing. *Requirements Engineering* 10, 198–211 (2005)
- [18] Bunke, H., Shearer, K.: A Graph Distance Metric Based on the Maximal Common Subgraph. *Pattern Recognition Letters* 19, 255–259 (1998)
- [19] van Dongen, B.F., Dijkman, R., Mendling, J.: Measuring similarity between business process models. In: Bellahsene, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 450–464. Springer, Heidelberg (2008)
- [20] Nejati, S., Sabetzadeh, M., Chechik, M., Easterbrook, S., Zave, P.: Matching and merging of statecharts specifications. In: Proc. of ICSE 2007, pp. 54–63 (2007)

- [21] Zhang, K., Shasha, D.: Simple Fast Algorithms for the Editing Distance between Trees and Related Problems. *SIAM Journal of Computing* 18(6), 1245–1262 (1989)
- [22] Sung, W.-K.: *Algorithms in Bioinformatics*. Chapman&Hall/CRC (2010)
- [23] Esgin, E., Senkul, P.: Delta Analysis: A Hybrid Quantitative Approach for Measuring Discrepancies between Business Process Models. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) *H AIS 2011, Part I. LNCS*, vol. 6678, pp. 296–304. Springer, Heidelberg (2011)

Qualitative Reasoning on Complex Systems from Observations^{*}

Gonzalo A. Aranda-Corral¹, Joaquín Borrego-Díaz², and Juan Galán-Páez²

¹ Universidad de Huelva. Department of Information Technology.
Crta. Palos de La Frontera s/n. 21819 Palos de La Frontera. Spain

² Universidad de Sevilla. Department of Computer Science and Artificial Intelligence.
Avda. Reina Mercedes s/n. 41012 Sevilla. Spain

Abstract. A hybrid approach to phenomenological reconstruction of Complex Systems (CS), using Formal Concept Analysis (FCA) as main tool for conceptual data mining, is proposed. To illustrate the method, a classic CS is selected (cellular automata), to show how FCA can assist to predict CS evolution under different conceptual descriptions (from different observable features of the CS).

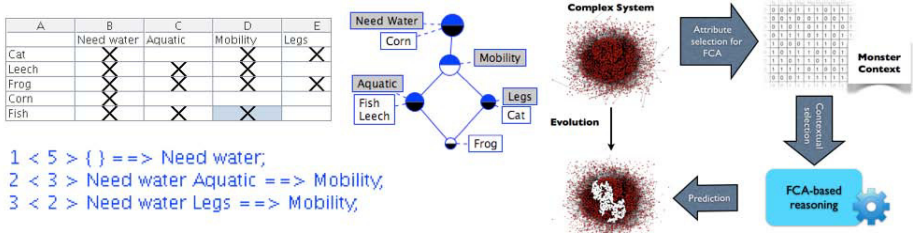
1 Introduction

The task of understanding a phenomenon amounts to find a reasonably precise and concise approximation to this phenomenon and its behavior such that it can be grasped by the human brain. New methods and tools have to be developed in order to assist experimental design and interpretation for: Identifying relevant entities at a given time and space scale, characterizing interactions between entities, and finally assessing and formalizing the system behavior [7].

Formal epistemology can play a relevant role. An adequate selection of key features and their dynamics specification is the first step in order to reconstruct the phenomena. In multilevel CS, the selection task requires a complex analysis of the different abstraction layers and organization levels. In classical systems as Cellular Automata (CA), the selection is limited by geometric and topological constraints so it could be more feasible. Human observation of CA allows to conjecture simple rules about the local dynamics, in order to explain the system dynamics as well as to isolate key concepts to forecast its evolution. Formal Concept Analysis (FCA) [8] provides tools and methods for extracting semantic features from data. FCA is a mathematical theory for data analysis, using formal contexts and concept lattices as key tools.

The aim of this paper is twofold. On the one hand, to show how FCA is used in the phenomenological reconstruction of CS dynamics, of qualitative nature. On the other hand it also aims to show how the selection of observable features influences the reconstruction, particularly the attributes on objects and interactions. To exemplify this idea, a well-known example, Conway's game of life (GoL), has been selected as running example, although the methodology is applicable to a wide class of CA.

^{*} Supported by TIC-6064 Excellence project (*Junta de Andalucía*) cofinanced with FEDER funds.



The next section reviews the basic elements of FCA, focusing on the use of implication basis (and association rules) for reasoning with formal contexts as basic data structure for qualitative observations. Sect. 3 succinctly presents contextual selection reasoning. In Sect. 4 GoL is used to show how CA is modeled by means of FCA which is also applied to a probabilistic Conway’s CA variant (Sect. 5). Sect. 6 is devoted to conclusions of the work and related work.

2 Background: Formal Concept Analysis

According to R. Wille, FCA mathematizes the philosophical understanding of a concept as a unit of thoughts composed of two parts: the extent and the intent. The extent covers all objects belonging to this concept, while the intent comprises all common attributes valid for all the objects under consideration. In this section, we succinctly present basic FCA elements (see [8] for a detailed exposition).

A formal context $M = (O, A, I)$ consists of two sets, O (objects) and A (attributes) and a relation $I \subseteq O \times A$. Finite contexts can be represented by a 1-0-table (identifying I with a boolean function on $O \times A$). See Fig. 1 top-left. The main goal in FCA is to compute the *concept lattice* extracted from the context. Given $X \subseteq O, Y \subseteq A$ it defines

$$X' := \{a \in A \mid oIa \text{ for all } o \in X\} \text{ and } Y' := \{o \in O \mid oIa \text{ for all } a \in Y\}$$

A (formal) concept is a pair (X, Y) such that $X' = Y$ and $Y' = X$. For example, concepts from formal context about living beings (Fig. 1, center) are depicted as a lattice. Actually in this lattice, each node is a concept, and its intension (or extension) can be formed by the set of attributes (or objects) included along the path to the top (or bottom). For example the node tagged with the attribute *Legs* represents the concept $(\{Legs, Mobility, NeedWater\}, \{Cat, Frog\})$ (which could be interpreted as the concept *land animal* in this context).

Knowledge Bases (KB) in FCA are formed by *implications between attributes*. An implication is a pair of sets of attributes, written as $Y_1 \rightarrow Y_2$. It is true with respect to $M = (O, A, I)$ according to the following definition. A subset $T \subseteq A$ respects $Y_1 \rightarrow Y_2$ if $Y_1 \not\subseteq T$ or $Y_2 \subseteq T$. $Y_1 \rightarrow Y_2$ is said to hold in M ($M \models Y_1 \rightarrow Y_2$ or $Y_1 \rightarrow Y_2$ is an implication of M) if for all $o \in O$, the set $\{o\}'$ respects $Y_1 \rightarrow Y_2$.

Definition 2.1. Let \mathcal{L} be a set of implications and L be an implication.

1. L follows from \mathcal{L} ($\mathcal{L} \models L$) if each subset of A respecting \mathcal{L} also respects L .
2. \mathcal{L} is complete if every implication of the context follows from \mathcal{L} .
3. \mathcal{L} is non-redundant if for each $L \in \mathcal{L}$, $\mathcal{L} \setminus \{L\} \not\models L$.
4. \mathcal{L} is a (implication) basis for M if \mathcal{L} is complete and non-redundant.

A particular basis is the so called *Stem Basis* (SB) [9]. SB for the context of Fig. 1 is shown (down). In this paper no specific property of the SB is used, so it can be replaced by any other. In order to reason with implications, a production system can be used [3].

Theorem 1. *Let \mathcal{S} be a basis for M and $\{A_1, \dots, A_n\} \cup Y \subseteq A$. The following statements are equivalent:*

1. $\mathcal{S} \cup \{A_1, \dots, A_n\} \vdash_p Y$ (\vdash_p is the entailment with the production system).
2. $\mathcal{S} \models \{A_1, \dots, A_n\} \rightarrow Y$
3. $M \models \{A_1, \dots, A_n\} \rightarrow Y$.

Implication basis are designed for entailing true implications only. When working on predictions Theorem 1 does not provide a sound method. In this case it is better to consider association rules (with confidence) from the *Luxenburger Basis* [17] instead of SB. The production system must be revised for working with confidence [4].

In FCA, association rules are also implications between sets of attributes. Confidence and support are defined as usual in data mining. The *Stem Kernel Basis* (SKB) is the subset of the SB formed by the implications with nonzero support. SKB are useful in a number of applications (cf. [3,2]).

3 Bounded (Automated) Reasoning on Complex Systems

The general approach to FCA-based qualitative reasoning on CS is based on considering local interaction as objects, which have several (local, observable) features (attributes) (see Fig. 1 right). Once the observer selects the attributes to be studied on the system, He/she can consider local interactions or nodes as objects of a formal context. This context \mathbb{M} (often a huge formal context) is built by means of data extraction, database processing, expert observations, data mining, etc. The observer has to select attributes and objects he considers relevant to determine CS dynamics, and the reasoning focuses on the associated subcontext (contextual selection). It is expected that reasoning with the contextual selection gives some information about the CS. In [4] this approach was applied using argumentative reasoning on contextual selections.

Particularly interesting is the case of predicting events when \mathbb{M} represents attributes on past events. The inference process consists of three steps [4]:

1. A question raises on whether a new event (object) has a property (attribute). Some properties on the new object are known (attribute values) $\{A_1, \dots, A_n\}$.
2. Selection provides a relatively small set of attributes, selected from own experience and beliefs, which are relevant on the object (according to observer's opinion).
3. The production system is executed on $\mathcal{L} \cup \{A_1, \dots, A_n\}$, where \mathcal{L} is a basis for the context induced by attribute selection made in step 2. The results obtained are the attributes inferred about the new object.
If the attribute B is inferred by the production system, then B is conjectured on the object.

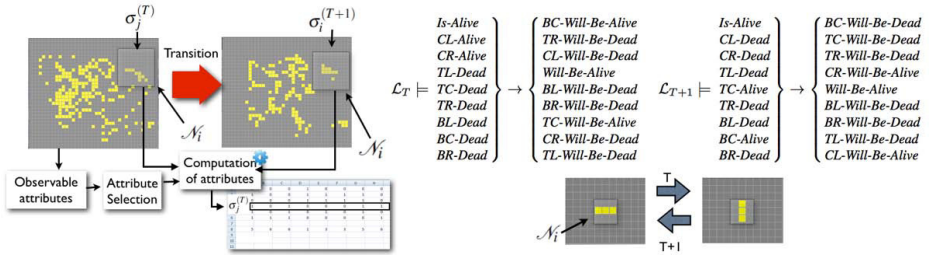


Fig. 2. Modeling CA with FCA (left). An oscillator with its production system (right)

Association rules are extracted from the contexts and used by the production system. From these association rules and some initial facts, based on the event we want to predict, the production system infers the confidence (probability) for each one of the possible values for unknown attributes on the event. Thus attributes constitute one of the most important and sensitive parts of the system. Lastly, attribute selection allows isolating a piece of \mathbb{M} where argument reasoning is based.

3.1 Describing CA Dynamics by Means of FCA

The aim of the paper is to show how this method provides a phenomenological reconstruction of CA, a CS where information flow is closed, namely Conway’s Game of Life (GoL). Whilst in [4] the method is applied to a CS with external information flow. Thus It was not possible to validate the method beyond purely experimental considerations.

In classic CA the new state of a cell only depends on the neighborhood configuration at the preceding time step (even it is also possible to consider memory capabilities [1]). If σ_i^T denotes the value of cell i at time step T , the evolution is an iteration of a mapping

$$\sigma_i^{(T+1)} = \phi(\{\sigma_j^{(T)} : j \in \mathcal{N}_i\})$$

where ϕ is an arbitrary function which specifies the cellular automaton rule operating on the cells in the neighborhood \mathcal{N}_i of the cell i . The standard framework of CA can be extended by implementing memory capabilities in cells: $\sigma_i^{(T+1)} = \phi(\{s_j^{(T)} : j \in \mathcal{N}_i\})$ with $s_j^{(T)}$ being a state function of the series of states of the cell j up to time-step T ; $s_j^{(T)} = s(\sigma_j^{(1)}, \dots, \sigma_j^{(T)})$.

The aim is to compute ϕ from observable features of \mathcal{N}_i (attributes) by means of FCA reasoning. Let $\mathbb{M} = (\mathbb{O}, \mathbb{A}, \mathbb{I})$ be the formal context whose objects \mathbb{O} are cells, and attributes \mathbb{A} are (computable) boolean properties (relation \mathbb{I} between objects and attributes) on the cells (for example, *Is-Alive*), considering, if it is necessary, past time steps (see Fig. 2, left).

From this formal context, SB, SKB and association rules can be computed. These are the Knowledge Basis (KB) containing a full or partial representation of CA dynamics, to be used in the reasoning process (i.e. to predict the evolution of a CA when its rules are unknown). Also the attribute selection (that is, the selection of spatio-temporal features on cells, the observer thinks that are relevant to decide the future state) is a key step, and the logical complexity of the description depends on this selection.

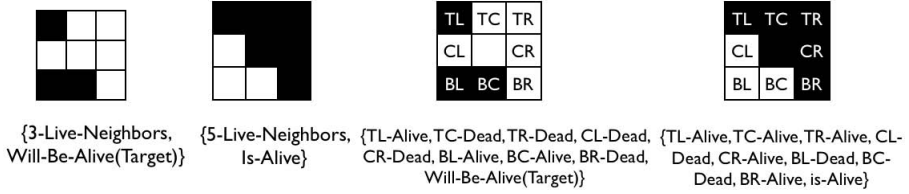


Fig. 3. Attributes for cells using *N-Neighbors* (left) and *Geometric* representation (right)

4 Modeling Game of Life by Means of FCA

In order to show how CA can be analyzed with the above described method, Conway’s Game of Life (GoL) (popularized by M. Gardner [10]) has been selected as running example, both the original one and an stochastic version.

Attribute Selection. The framework is similar to when an observer aims to predict the future state of CA from the observation of its evolution after a number of transitions. Two steps are needed: 1) Choose topological/geometrical properties which are considered relevant to describe system’s evolution. 2) Conjecture, based on these properties, the rules governing the system. Two ways of describing current cell’s environment are considered, which correspond to two ways of feature selection by the observer:

Attributes based on the number of alive neighbors (N-Neighbors): This modeling is specific for GoL, as it is based on the number of alive neighbors. The attribute set has 11 attributes (See Fig. 3 left for an example): 9 Attributes describing the neighborhood: $\{0\text{-Live-Neighbors}, \dots, 8\text{-Live-Neighbors}\}$, one attribute describing current cell state: *Is-Alive*, and one attribute describing the cell state in the next generation: *Will-Be-Alive(Target)* (which is the target attribute in the reasoning process).

Attributes based on each neighbor state (Geometric): This modeling is not specific for GoL, but is robust enough to be used with many diverse CA. The state of each neighbor is specified individually, considering the Moore neighborhood. This attribute set consists on 18 attributes (see Fig. 3, right): 16 Attributes specifying (geometrically) whether each neighbor is alive or dead: $\{Top\text{-Left-Alive}, Top\text{-Left-Dead}, \dots, Bottom\text{-Right-Alive}, Bottom\text{-Right-Dead}\}$ and the attributes *Is-Alive* and *Will-Be-Alive(Target)* as in the above representation.

FCA Based Reasoning for CA. Once $\mathbb{M} = (\mathbb{O}, \mathbb{A}, \mathbb{I})$ is built (as above described), the concept lattice (see Fig. 4, top) and SB, \mathcal{L}_{GoL} , are computed. In the case of N-Neighbors representation, It matches Conway’s rules. This implicational basis (\mathcal{L}_{GoL}) is the aforementioned KB. In Fig. 4 the meaning of a concrete rule (from the KB obtained using the *N-Neighbors* representation) is explained. Note that the concept *cell that survives* is extracted from the formal context. In fact, the following holds,

$$\mathcal{L}_{GoL} \models 2\text{-live-Neighbors} \rightarrow (Is\text{-Alive} \leftrightarrow Will\text{-Be-Alive}(Target))$$

which gives some insights on live persistence in GoL.

Preliminary experiments showed that both representations described suffices for predicting GoL behavior. Using just one transition as KB, from time step $N - 1$ to N , it is possible to predict CA state in the time step $N + 1$. SB using the *Geometric*

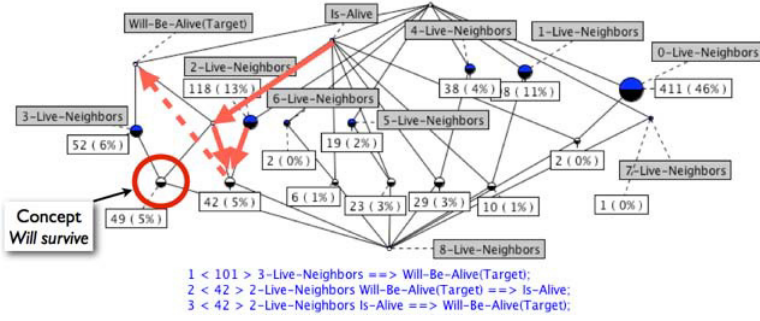


Fig. 4. Concept lattice (N -Neighbors) showing rule 3 (top) of its SB (bottom)

representation has a considerably bigger size (more than 700 rules) than the SB using the N -Neighbors representation (3 rules). For instance, the recognition of persistence or oscillatory objects in GoL depends on FCA modeling and historic features of CA. For example, to recognize that the blinker (Fig. 2, bottom right) is an oscillator with period 2, it suffices to prove the two facts shown in Fig. 2 (top right) with the production system (\mathcal{L}_j is the SB for the transition step j).

Soundness: The method on a bounded region of CA universe, depends on both the attribute selection and the size of the region observed. In the case of CA in which ϕ uses N -neighbors or Geometric attributes, ϕ induces a set of implications on attributes, denoted by S_ϕ , defined as follows. For the sake of simplicity, only geometric attributes are considered (the other case is similar). Let L_N the set of configurations of the neighborhood \mathcal{N} on which ϕ takes the value *Will-be-Alive* and let D_N its complement (on which ϕ outputs *Will-be-Dead*). It is described by means of two formulas

$$\bigvee_{C \in L_N} C \rightarrow \text{Will-be-Alive}, \text{ and } \bigvee_{E \in D_N} E \rightarrow \text{Will-be-Dead}$$

where each C, E is a conjunction. Let be $S_\phi = \{C \rightarrow \text{Will-be-Alive} : C \in L_N\}$ and $N_\phi = \{E \rightarrow \text{Will-be-Dead} : E \in D_N\}$. Note that S_ϕ characterizes ϕ , so it can be considered as an equivalent characterization. In this case, the attribute *Will-be-Dead* should be also considered in the representation. The soundness is stated as follows (ahistoric CA, Moore neighborhood):

Theorem 2. *Let be a CA specified by a function ϕ . For any nontrivial initial population density δ (that is, $0 < \delta < 1$ being the probability for a cell to be alive) it has*

$$\lim_M \text{Prob}(\mathcal{K}_M \models S_\phi \cup N_\phi) = 1$$

where \mathcal{K}_M is the Stem Kernel Basis for the the first transition of the CA restricted to the rectangle $I_M = (-M, M) \times (-M, M) \subset \mathbb{Z}^2$

Table 1. Experiments. $\#P$, $\#C_{TH}$ and $\#runs$ are the number of intervals for P, C_{TH} and number of runs resp. and SB average size for $C_{TH} = 1.0$) and $P = 1.0$

CA modeling	$\#P$	$\#C_{TH}$	N_{exec}	$\#runs$	SB average
<i>N-Neighbors</i>	100	100	50	500,000	3.25
<i>Geometric</i>	50	50	7	17,500	735.91

5 Cellular Automata with Probabilistic Features

The framework is extended to probabilistic CA's for dealing with real world situations, where rules are unknown and the information available comes from observations.

In the probabilistic CA, in each time step, P is the probability for each cell to behave normally, and $1 - P$ the probability to behave randomly. Since SB do not consider any rule exception, in order to deal with probabilistic CA (uncertain reasoning), it is more appropriate to choose association rules. Thus the production system used in this case is a bit different to the one mentioned before (it works like in [4]). As the confidence of association rules measures the truth degree of the rules within the context, a confidence threshold (C_{TH}) is selected to choose a rule subset as KB in the reasoning process.

5.1 Experiments

The goal of the experiments is to test the reliability of FCA-based reasoning for simulating CA dynamics. To this aim, one experiment for each of the two representations of CA is presented, in order to test the accuracy (measuring the *error rate*) of the reasoning system for different values of C_{TH} and P .

Some parameters should be selected to set up the experimentation environment. 1) *grid size* (1000 cells). 2) *Initial grid density* (around 30% of alive cells¹). 3) The transition used to build the KB (Gen_{KB} . From generation 1 to 2). 4) The generation to be predicted by the reasoning system (Gen_{query} . Generation 3). Finally, three dimensions were considered in order to perform different experiments and explore the results: 1) Confidence threshold C_{TH} for the KB, 2) Probability P for the probabilistic GoL and 3) Number of cells the system could not predict properly (*error rate*). For each different value of C_{TH} and P the system is executed N_{exec} times in order to obtain the average *error rate* in the prediction of the next state. Each execution is as follows:

1. CA grid is randomly initialized with a fixed initial density.
2. A first transition of the CA is simulated (with probability P) to obtain Gen_{KB} .
3. A formal context $\mathbb{M} = (\mathbb{O}, \mathbb{A}, \mathbb{I})$ is built with the information of Gen_{KB} .
4. Extraction of association rules set, using the threshold C_{TH} to obtain the KB
5. For each cell an attribute set with the description of its neighborhood state in the 2nd generation is computed. The system is executed on these attributes, to infer whether the cell will be alive or dead in the 3rd generation, according to the selected criterion.
6. The error rate is measured.

¹ We have selected this initial demographic density for probabilistic experiments because the experiments showed long non-stable behaviors for most of P values.

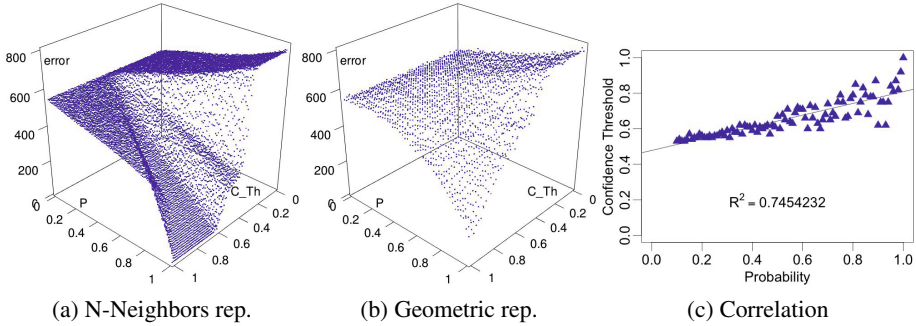


Fig. 5. Experimental results ((a), (b)) and Correlation between C_{TH} and P minimizing *error rate* for *N-Neighbors* representation

5.2 Discussion/Results

Results of experiments are shown in Fig. 5 (plots a and b). As $1 - P$ is the probability of a random state change to occur in a cell, the *error rate* is expected to grow fast. When $P < 0.5$, CA behavior tends to be chaotic and unpredictable.

It is interesting to observe how the uncertainty introduced by P is countered by the confidence of association rules. The values of P and C_{TH} are correlated in order to minimize the *error rate*. Fig. 5 (a, b) shows how the confidence threshold C_{TH} decreases linearly with the probability P in order to minimize the *error rate* (recall that randomness grows as P decreases). It is worth to note the case of *N-Neighbors* where the points minimizing the *error rate* form an almost perfect line when $P \geq 0.1$. When $P < 0.1$ the behavior of the CA is fully random. Fig. 5 (plot c) shows for each probability value $P \geq 0.1$, the value of C_{TH} minimizing the *error rate*. Also the Pearson’s correlation coefficient for the same dataset is 0.863. This high correlation between the probability (uncertainty) of the CA and the confidence threshold used to select the rules set of the reasoning system shows system’s resistance against noise and randomness.

Results show that *geometric* representation is less accurate (but acceptable). If we think in the fast and frugal way [12], this will be the representation to be used with any kind of CA based on the Moore neighborhood, as it is more robust when modeling any unknown problem. Finally, it is interesting to remark, that the huge difference between both representations in the SB size (Table 1) will not suppose a big difference in computation time. The reasoning system works with logical implication between attributes, thus once the implication basis has been computed, the execution of the reasoning system is quite light. Moreover, the Stem Basis is minimal, therefore SB size will be similar in any experiment where the size of the considered grid is big enough.

6 Conclusions and Related Work

The methodology for short-term prediction of CS evolution, previously used in [4], allows to outline the relationship between the system's features (attributes) choice and the complexity of the logical description of its evolution (by means of FCA). We have selected as running example the well-know GoL system, but the methodology can be applied to other more complex examples. In [4] it is shown how a sound attribute selection can make this prediction method better than classic learning systems. However, in [4], the soundness of the method is justified only in experimental terms, due to the fact that an information flow, external to the system, already exists. In the case of this work, the CS under study has a *closed environment* (without external information flow). Thus it has been shown that not only the method works, but also its correctness, in asymptotic terms, can be demonstrated. This method constitutes a hybrid method due to the fact that FCA does not consider non-deterministic reasoning.

With respect to FCA-like approaches for mining dynamics of systems, in [6] a notion of (deterministic) association rule for ordered data is proposed, proving that the result can be formally justified by using background knowledge, and FCA is applied on ordered contexts. Implications can be considered as a specialized Horn-like propositional clauses. Therefore, logical machinery designed for Horn logic reasoning can enrich the framework presented in this paper, particularly those which work with ordered data [5,6]. This question is the aim of a future work.

With respect to CA field, the method produces logical representations of transitions which may be related with λ parameter [13]. It could be useful to analyze the behavior of Stem basis in probabilistic versions of CA [22,15]. A FCA-based formalization for CA with memory can be a descriptive system on which validate specific conditions in future formal methods to specify emergence in CA [21]. Moreover, FCA also provides a strong relation between implications and the context. Classifications as that of given in [14] can be an interesting starting point to extend FCA with asymptotic studies on formal contexts which evolve. In [16] it is also shown a method for data mining of CA transition rules focused on geographic applications. In [20] the authors use genetic algorithms in the learning phase whilst we could offer a logical argument of the learning of rules.

The probability δ for the initial population of experiments is a key parameter in order to determine the evolution of the CA, thus the above mentioned limit strongly depends on its value. A similar (although more specific) question is studied in [11], where the existence of CA with fixed point configurations depending on initial density are considered. From the point of view of our paper, we could say that certain implicational description of the world has the formal context associated to the current state as fixed point. Entropy features of formal contexts should be considered, in order to relate FCA representation with asymptotic behavior of CA [19], as well as to extract conceptual structure in CA with similar behavior without human engineering [18].

Classical learning process does not provide a straightforward method to discover new geometrical concepts, as FCA can do (identifying some formal concepts as the FCA-based definition of stable colonies or gliders, by analyzing their extents) by expanding attribute set and Moore neighborhood (even by using geometric -non isotropic- attributes) as well as attributes with bounded temporal stamps [1].

References

1. Alonso-Sanz, R.: LIFE with Short-Term Memory. In: Adamatzky, A. (ed.) *Game of Life Cellular Automata*, pp. 275–290. Springer (2010)
2. Aranda-Corral, G.A., Borrego-Díaz, J., Giráldez-Cru, J.: Agent-mediated shared conceptualizations in tagging services. *J. Multimedia Tools and Applications* 65(1), 5–28 (2013)
3. Aranda-Corral, G.A., Borrego-Díaz, J.: Reconciling Knowledge in Social Tagging Web Services. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) *HAIS 2010, Part II. LNCS*, vol. 6077, pp. 383–390. Springer, Heidelberg (2010)
4. Aranda-Corral, G.A., Borrego-Díaz, J., Galán-Páez, J.: Complex Concept Lattices for Simulating Human Prediction in Sport. *J. Syst. Science and Complexity* 26(1), 117–136 (2013)
5. Balcázar, J.L., Garriga, G.C., Díaz-López, P.: Reconstructing the rules of 1D cellular automata using closure systems. In: *Proceedings of the 2nd European Conference on Complex Systems*, pp. 55–61 (2005)
6. Balcázar, J.L., Garriga, G.C.: Horn axiomatizations for sequential data. *Theor. Comput. Sci.* 371(3), 247–264 (2007)
7. Bourguine, P., Chavalarías, D., Perrier, E. (eds.): *CSS Roadmap for the Science of Complex Systems* (2009)
8. Ganter, B., Wille, R.: *Formal Concept Analysis. Mathematical Foundations*. Springer (1999)
9. Guigues, J.-L., Duquenne, V.: Familles minimales d' implications informatives resultant d'un tableau de donnees binaires. *Math. Sci. Humaines* 95, 5–18 (1986)
10. Gardner, M.: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American* 223, 120–123 (1970)
11. Gog, A., Chira, C.: Cellular Automata Rule Detection Using Circular Asynchronous Evolutionary Search. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) *HAIS 2009. LNCS*, vol. 5572, pp. 261–268. Springer, Heidelberg (2009)
12. Goldstein, D., Gigerenzer, G.: Fast and frugal forecasting. *Int. J. Forecasting* 25, 760–772 (2009)
13. Langton, C.G.: Computation at the edge of chaos: phase transitions and emergent computation. In: *Proc. 9th Int. Conf. Cent. Nonlinear Studies on Self-organizing, Collective, and Coop. Phen. in Nat. and Art. Comp. Networks on Emergent Computation*, pp. 12–37 (1990)
14. Li, W., Packard, N.: The Structure of the Elementary Cellular Automata Rule Space. *Complex Systems* 4, 281–297 (1990)
15. Li, W., Packard, N.H., Langton, C.G.: Transition phenomena in cellular automata rule space. In: *Cellular Automata*, pp. 77–94. MIT Press (1991)
16. Li, X., Yeh, A.G.-O.: Data mining of cellular automata's transition rules. *Int. J. Geograp. Inf. Sci.* 18(8), 723–744 (2004)
17. Luxenburger, M.: Implications partielles dans un contexte. *Math. Inf. Sci. Hum.* 11, 335–355 (1991)
18. Marques, M., Manurung, R., Pain, H.: Conceptual representations: What do they have to say about the density classification task by CA? In: *Proc. 2006. Eur. Conf. on Complex Systems* (2006)
19. Packard, N., Wolfram, S.: 2-Dimensional Cellular Automata. *J. Stat. Physics* 38(5-6), 901–946 (1985)
20. Piwonska, A., Seredynski, F.: Learning cellular automata rules for pattern reconstruction task. In: Deb, K., et al. (eds.) *SEAL 2010. LNCS*, vol. 6457, pp. 240–249. Springer, Heidelberg (2010)
21. Sanders, J.W., Smith, G.: Emergence and Refinement. *Formal Aspects Comp.* 24(1), 45–65 (2012)
22. Wootters, W.W., Langton, C.G.: Is there a sharp phase transition for deterministic cellular automata? *Phys. D* 45(1-3), 95–104 (1990)

Reference Data Sets for Spam Detection: Creation, Analysis, Propagation

Marcin Luckner¹ and Robert Filasiak²

¹ Warsaw University of Technology,
Faculty of Mathematics and Information Science,
pl. Politechniki 1, 00-661 Warsaw, Poland
mluckner@mini.pw.edu.pl

<http://www.mini.pw.edu.pl/~lucknerm/en/>

² Orange Labs Poland
02-691 Warszawa, ul. Obrzeza 7, Poland
Robert.Filasiak@orange.com

Abstract. A reference set is a set of data of network traffic whose form and content allows detecting an event or a group of events. Realistic and representative datasets based on real traffic can improve research in the fields of intruders and anomaly detection. Creating reference sets tackles a number of issues such as the collection and storage of large volumes of data, the privacy of information and the relevance of collected events. Moreover, rare events are hard to analyse among background traffic and need specialist detection tools. One of the common problems that can be detected in network traffic is spam. This paper presents the methodology for creating a network traffic reference set for spam detection. The methodology concerns the selection of significant features, the collection and storage of data, the analysis of the collected data, the enrichment of the data with additional events and the propagation of the set. Moreover, a hybrid classifier that detects spam on relatively high level is presented.

Keywords: Reference sets, Spam detection, Flow analysis, Anomaly detection, Hybrid classifiers.

1 Introduction

A reference set is a set of data of network traffic whose form and content allows detecting a chosen event or a group of events. Such set is a subset large enough to allow an analysis of an event but significantly smaller than the source traffic. Realistic and representative datasets can improve research into anomalies detection, attack prevention, and botnet detection.

However, the creation of reference sets faces serious problems such as the collection and storage of large volumes of data, the privacy respect, and the relevance of collected events. Therefore, there is lack of useful reference sets [4].

Moreover, some events are rarely present in network traffic. Therefore, their detection needs special analytic tools.

Our work focuses on spam detection. It was proved that spam could be detected on the base of TCP/IP features. Proposals of such solutions are given in [10,12,11]. However, our aim is not only to detect spam but also to create a complete solution that allows making reference sets from data collected in the future.

This paper presents the methodology of network datasets. The methodology concerns such aspects of the issue as the selection of significant features that describe the observed events, data collection and storage, an analysis of the collected data for the presence of the events, the enrichment of the data with additional events, and the propagation of the set.

2 Reference Set Creation

The creation of a reference set consists of the following stages. The first stage is the selection of parameters. The subset of parameters (m_1, \dots, m_n) from a wide domain of characteristics M is selected for observation (Section 2.1). Next, values of the selected parameters are captured from real data. Based on the captured values, features are calculated (Section 2.2). Finally, decision rules are created. In this work, rules are defined by random forests and support vector machines (Section 2.3).

Two additional conditions should be fulfilled. First, the set must contain sufficient information about events (Section 2.4). Second, the set should be transformed to a form that can be accepted by a diagnostic unit (Section 2.5).

In the following sections, all stages are presented on the example of a reference set that contains spam.

2.1 Features Selection

For significant feature detection, it is necessary to define the range of collected data. As a source of features, a set of low-level features described in [9] may be used. The proposed set contains such simple features as a package length, TCP window size, or TTL. Based on the features, a number of statistics such as count, minimum, maximum, and average are calculated.

To perform an analysis, two distinguishable sets are necessary. The first set contains the analysed events and the second consists of the rest of network traffic. Both sets are described by a subset of low-level features. Feature selection creates a subset of features that discriminate events from the background traffic.

The selection is realised by a decision tree. In the tree creation process, the Gini coefficient is calculated and used as the measure of discrimination ability for selected features [1].

The tree creates a set of rules that can be used for approximate separation of events from background traffic. However, such a classifier may be insufficient for complex tasks.

In the case of spam, flow-level parameters $\{m\}$ selected by Žádník [12] as a subset of the set $\{M\}$ defined in [9] are the base for features selection. The features

Table 1. The most significant features describing spam

Significance	Name	Description	Statistic	Subject
100,0	slas	Average length of package having the ACK flag	Average	ACK
99,5	spl	Average package length	Average	Package
91,0	slps	Average length of package having the PUSH flag	Average	PSH
90,5	maxpl	Maximum package length	Maximum	Package
84,0	maxtw	Maximum TCP window size	Maximum	Window
82,5	spps	Count of packages having the PUSH flag	Count	PSH
73,0	stw	Average TCP window size	Average	Window
68,0	mintw	Minimum TCP window size	Minimum	Window
67,5	maxttl	Maximum TTL	Maximum	TTL
67,0	sp	Packages count	Count	Package
66,5	minttl	Minimum TTL	Minimum	TTL
66,5	sttl	Average TTL	Average	TTL

were calculated for flows collected by Žádník and Michlovský [12]. The authors collected data from the SMTP server hosting mailboxes of the LiberoRouter¹ project group. The data set contains over 58 thousand records described by 64 features and divided into several classes. Among all classes, two describe spam. The first class *dnsbl* contains flows from IP address mentioned on DNS black lists². The second class *y_spam* consists of flows that were successfully received and marked as spam by SpamAssassin³. In this paper, both classes are considered as a single class *spam*.

The same features set was used in a new set of NetFlow records that was collected at Warsaw University of Technology. The set originates from the mail server Alpha and consists of NetFlow records described by the same collection of features as the Žádník's set. The data was collected over one working week. More than 42 thousand NetFlow records were collected. Among them 589 were labelled as spam.

It was proved that the set of 64 features proposed by Žádník contain unnecessary features and the dimension of the classification task can be reduced [5]. Therefore, additional features selection is necessary.

For both sets, the discrimination rules that separate spam from the rest of the traffic were created. This task was done using a C&RT tree [2].

The accuracy of both classifiers was about 97 percent. However, the tree structures were different. Therefore, various features were selected as the most significant. The final features set should not be based only on the features selected as major by both classifiers because such set would be very limited. Instead, the set can be extended by the features selected by only one classifier.

The most significant features are presented in Table 1. Information about the direction of traffic is skipped. It is assumed that the mentioned features should be calculated for both directions. That gives 24 features.

¹ <http://liberouter.org/>

² <http://cbl/abuseat/org/>

³ <http://spamassassin.apache.org/>

2.2 Flow Collection

Two approaches seem to be interesting from the perspective of multi-gigabit stream analysis: packet header analysis and flow analysis. Both of them do not utilise the information contained in payload. This fact is very important for data volume reduction and privacy. Additionally, a hash function can be used to obfuscate the IP addresses and guarantee privacy. In some cases, data does not need to be stored. A good example is the statistical detection of DDoS (Distributed Denial of Service) attacks [8] and the solutions developed on FPGA (Field Programmable Gate Array) cards [7].

The packet header analysis is focused on packet headers. The flow analysis is focused on sets of header determined by the source and destination IP, the source and destination port, timestamps, etc. depending on the parameters used to define the flow. The flow analysis enables more compact data reduction. IPFIX (IP Flow Information Export) and NetFlow are standards. The equivalent given by Juniper is named J-flow.

Many aspects of network traffic can be described using NetFlow v9 [6]. This protocol allows the users to define their own fields. NetFlow records from PCAP frames can be calculated by nProbe that collects traffic data and emits NetFlow v9 flows towards a specified collector [3].

The size of collected flows is less than that of complete frames or even just headers. However, even for a very short period, the collection of data from a BRAS (Broadband Remote Access Server) creates multigigabit files. Therefore, only a limited subset of the traffic can be analysed. There are two approaches to creating the subset. In the first approach, the subset is a cross section of the traffic. In this case, a random pool of clients IP addresses is selected and observed. The pool must maintain typical proportion between various types of clients (business, individual). In the second approach, the subset is a probe of typical environment where events occur. Such environment can be defined by some protocol and ports. The approaches can be combined.

Filtering rules for IP pool as well as for a protocol and ports can be described and applied during the collection process in the form of Berkeley Packet Filter, which is supported by nProbe.

Capture NetFlow Records. NetFlows were collected by nProbe⁴. This tool allows user to define a template of a NetFlow record. In the case of spam, the following template that contains elements mentioned in Table 2 was used.

The traffic was filtered using the BPC *Berkeley Packet Filter* rule that limits the collected data to the mail traffic:

(tcp and (dst port 25 or 465 or 587) or (src port 25 or 465 or 587))

These restrictions limit the collected traffic to the typical spam environment.

As the result, 176446 records were created. The total size of records was 17.5 MB. The limitation of the size is significant. For comparison, files that consist of PCAP frames without payload and include, after conversion into NetFlows, 406 STMP records achieve the size of 4.6 GB.

⁴ <http://www.ntop.org/products/nprobe/>

Table 2. Description of NetFlow parameters

Parameter	Description
<i>IPV4_SRC_ADDR</i>	IPv4 source address
<i>LA_SRC_PORT</i>	IPv4 source port
<i>IPV4_DST_ADDR</i>	IPv4 destination address
<i>LA_DST_PORT</i>	IPv4 destination port
<i>IN_BYTES</i>	Incoming flow bytes
<i>IN_PKTS</i>	Incoming flow packets
<i>PROTOCOL</i>	IP protocol
<i>TCP_FLAGS</i>	Cumulative of all flow TCP flags
<i>LAST_SWITCHED</i>	SysUptime (msec) of the last flow pkt
<i>FIRST_SWITCHED</i>	SysUptime (msec) of the first flow pkt
<i>OUT_BYTES</i>	Outgoing flow bytes
<i>OUT_PKTS</i>	Outgoing flow packets
<i>LONGEST_FLOW_PKT</i>	Longest packet (bytes) of the flow
<i>SHORTEST_FLOW_PKT</i>	Shortest packet (bytes) of the flow
<i>OOORDER_IN_PKTS</i>	Number of out of order TCP flow packets (from source)
<i>OOORDER_OUT_PKTS</i>	Number of out of order TCP flow packets (from destination)
<i>ICMP_TYPE</i>	ICMP type
<i>IN_SRC_MAC</i>	Source MAC Address

Features Calculation. The collected records, described in Table 2, should be rewritten to achieve a form similar to the set of the most important features that is presented in 1. The transformation is described in Table 3.

In the computation description, C notation is used. The operator *?* means *if* statement where preceding condition determines the returned value. If the condition is true, the value on the left side of the colon is returned. If the condition is false, the value on the right side of the colon is returned. The operator *%* denotes the modulo operation.

The created set of features consists of simple statistics calculated for package length and binary information about flags presence.

2.3 Probe Analysis

The probe collected from the whole traffic should be checked for the presence of the analysed events. This task can be done by a classifier trained to use a learning set. However, such classifier has a tendency to detect events described by the learning set while events from the probe may have different characteristics. Therefore, an ensemble of various classifiers trained on different data is a much better solution.

Each classifier from the ensemble recognises three classes: an event, background traffic and other. The last class contains all border cases. The classes are labelled by 1, -1, and 0 respectively. The classifier C_i that returns a decision y_i is described by two coefficients. The first one, s_i is the accuracy of discrimination between the events and the background. The second one, c_i is a confidence

Table 3. Computation of features from NetFlow parameters

Feature	Computation
Count of packages from source	<i>IN_PKTS</i>
Average length of package from source	<i>IN_BYTES/IN_PKTS</i>
Count of packages from destination	<i>OUT_PKTS</i>
Average length of package from destination	<i>OUT_BYTES/OUT_PKTS</i>
Minimum length of package	<i>SHORTEST_FLOW_PKT</i>
Maximum length of package	<i>LONGEST_FLOW_PKT</i>
Appearance of Urgent flag	<i>TCP_FLAGS%64 ≥ 32?1 : 0</i>
Appearance of Acknowledgement flag	<i>TCP_FLAGS%32 ≥ 16?1 : 0</i>
Appearance of Push flag	<i>TCP_FLAGS%16 ≥ 8?1 : 0</i>
Appearance of Reset flag	<i>TCP_FLAGS%8 ≥ 4?1 : 0</i>
Appearance of Synchronisation flag	<i>TCP_FLAGS%4 ≥ 2?1 : 0</i>
Appearance of Fin flag	<i>TCP_FLAGS%2 ≥ 1?1 : 0</i>

level calculated as a percent of decisions from the outside of the other class calculated as:

$$c_i = \frac{\sum_X |y_i|}{|X|}, \quad (1)$$

where X is the learning set, and y_i is a decision of the classifier.

The final classification decision is the sign of a weighted sum given by the formula:

$$y = \operatorname{sgn} \sum_{i=1}^n \frac{s_i}{\sum_{j=1}^n s_j} \frac{c_i}{\sum_{j=1}^n c_j} y_i, \quad (2)$$

where n is the number of classifiers in the ensemble.

Two classifiers were used to separate the traffic into three classes: spam, background traffic and others. The first one was support vector machine (SVM). The second one was a random forest. Both techniques were trained on records from the Žádník's set and Alpha separately. That gives four classifiers. Details about the accuracy and the number of false positives for spam are given in Table 4. For comparison, results obtained from the whole set of Žádník's features calculated for PCAP headers as well as results achieved from NetFlows features presented in Table 3 are given.

The result shows that classifiers based on NetFlow records are at least as good as those created from PCAP headers. There is no significant difference between both classification techniques used.

The probe collected from BRAS was analysed by an ensemble of three best classifiers. The group detected 834 spam records among 175515 background records. Additionally, eight others records were found. The ensemble was expanded by an additional classifier, eliminating the 'others' group and resulting in the detection of one more spam record.

Table 4. Comparison of classifiers

Classifier	Set	Accuracy		False positive	
		PCAP	NetFlow	PCAP	NetFlow
Random forests	Alpha	0.994	0.993	0.001	0.002
	Zádník	0.982	0.979	0.189	0.214
SVM	Alpha	0.990	0.990	0.002	0.003
	Zádník	0.848	0.939	0.057	0.063

The analysis shows the observed event: spam is almost imperceptible in the probe. If the probe were to be used as a reference set spam contamination should be added.

2.4 Probe Enrichment

If the number of records in the probe describing the analysed event is scarce, a new record should be added. The new records can come from learning sets or be generated on the base of existing records.

Contamination from outside sources assumes that the sources have the same set of features as the probe. If flows concern different pairs of sources and destinations, where the source and the destination is defined by the IP number and port, then a new set of flows can be simply concatenated with the probe. Otherwise, features should be recalculated. It is assumed that all features are simple statistics such as count, minimum, maximum, and average.

Contaminations may be also generated without outside influences using a genetic algorithm. Flows are described as sets of discrete and continuous features. For two randomly selected flows, a new pair of flows is created. Each new flow has discrete features of one parent. The continuous features are calculated as averages of parents' values.

Created Reference Sets. Apart from the collected probe, two additional reference sets were created. In both sets, the number of spam records was increased by 19 thousand. That gives about 10 percent of spam in the analysed probe, whereas in the original set it was about 0.5 percent.

The first set was created by adding records from an outside set. The records came from a different source and thus could be added by simple concatenation. The contamination changed the characteristic of spam in a significant degree.

The contamination of the second set was generated by the genetic algorithm. New records were generated on the base of the existing ones and the characteristic of spam has changed minimally.

The influence of both methods on spam characteristic is given in Fig. 1 where original averages of continuous features are presented alongside the values from the extended sets.

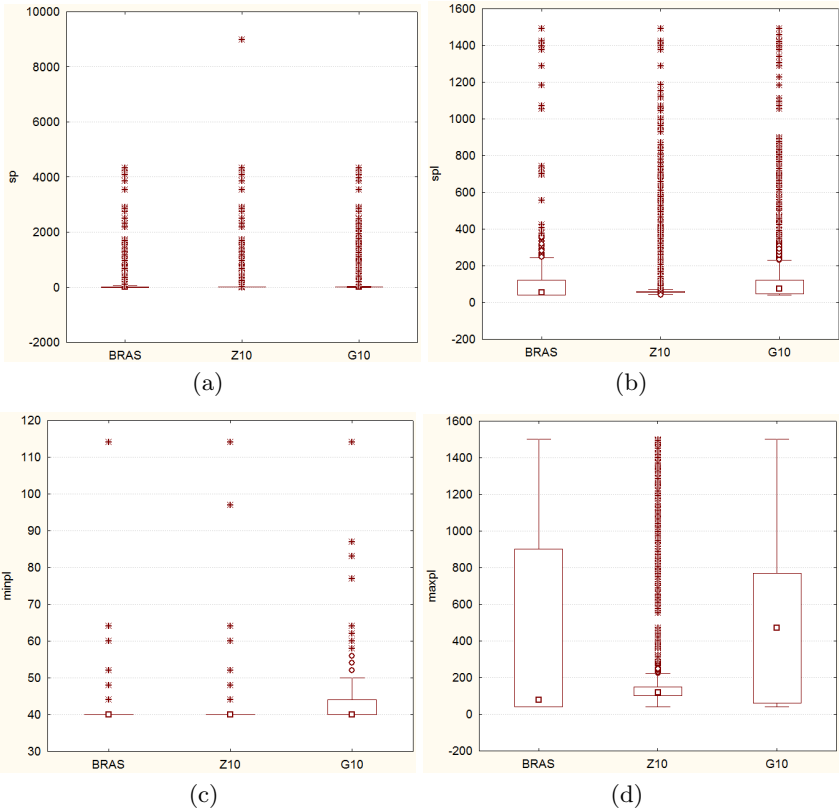


Fig. 1. Analysis of continuous features. Medians are marked with squares, boxes contain 50 percent of cases, and observations outside of the whiskers are outliers. Observations marked with stars are extremes. BRAS: Original spam; Z10: Original spam plus spam from the outside set; G10: Original spam plus spam generated by the genetic algorithm.

The most significant difference between the sets is on **maxpl** (Fig. 1(d)). The outside set has definitely different characteristic than the rest of sets. A similar situation occurs for the **spl** feature (Fig. 1(b)). Differences for the **sp** feature cannot be observed (Fig. 1(a)). An effect opposite to the intended occurs for the **minpl** feature where spam generated by the genetic algorithm has different characteristic but in this case, all records are positioned near the minimal size of TCP header (Fig. 1(c)).

2.5 Reference Set Propagation

The most important disadvantage of collecting data in the form of a NetFlow record is the problem of the reference set propagation. Diagnostic units accept input in the form of PCAP frames. The set consisting of NetFlow records must be converted into PCAP frames.

A single PCAP frame can be created by a packet infector such as *nemesis*⁵. However, a record from the set is an equivalent of a sequence of PCAP frames that fulfil statistic conditions such as the minimum, the maximum, the average and the count for a selected feature. The whole sequence can be created when two frames are created to fulfil the minimum *MIN* and the maximum *MAX* criteria and the rest of frames have a value of the feature calculated as the average modified by extremes.

In the discussed case, spam is defined by count and size of packages and the flags used. Therefore, it is enough to create two packages of the size *SHORTEST_FLOW* and *LONGEST_FLOW* respectively and the size of the rest of packages should be calculated as:

$$SIZE = \frac{IN_BYTES - LONGEST_FLOW - SHORTEST_FLOW}{IN_PKTS - 2}. \quad (3)$$

Where $IN_PKTS - 2$ is a number of packages generated in the given size. Finally, desired flags should be set.

3 Conclusions

This paper presents the methodology concerning several aspects of reference sets creation. The data was captured in the form of NetFlow records that contain features selected in the analysis based on a decision tree. The probe collected from the whole traffic should be checked for the presence of the analysed events. If necessary, the probe is enriched with additional records. The type of contamination created by the added records depends on the planned experiments. An outside set may be used when it described analysed events. If the probe contains the main aim of the research then a genetic algorithm should be used instead. The created set consists of NetFlow records. This form results in a significant reduction of size but cannot be used to test network devices. Therefore, a method is proposed that creates network traffic from NetFlow records.

The proposed methodology was used to create various reference sets for spam analysis. The collected traffic comes from a BRAS and was enriched by an outside set as well as by generated records. Various techniques were compared to present their suitability in different applications.

The presented process can be used to create reference sets for different events such as DDoS attacks, a netbots activity, or damage manifestations. The method can create new sets with relevant and current data with genetic operator created events that are a projection of the changes in the analysed data.

The proposed method has several significant advantages. First, only headers of messages are analysed. Therefore, a privacy of users is guaranteed. Second, the detection algorithm analyses only twelve simple features and results in 99 percent accuracy. Finally, the solution was verified on real data.

⁵ <http://nemesis.sourceforge.net/>

The created solution was used to detect spam among real data collected by an Internet service provider. Records from 64088 unique IP addresses were captured. Among the addresses, 359 have sent at least one record labelled spam. The significant part of them (46 percent) came from seven sources. Therefore, the described method allows an operator to detect sources of spam among typical network traffic.

References

1. Behera, G.: Privacy preserving c4.5 using gini index, pp. 1–4 (March 2011)
2. Breiman, L., Friedman, J., Olshen, R., Stone, C.: *Classification and Regression Trees*. Wadsworth and Brooks, Monterey (1984)
3. Deri, L.: nprobe: an open source netflow probe for gigabit networks. In: Proc. of Terena TNC 2003 (2003)
4. Fomenkov, M., Claffy, K.: Internet measurement data management challenges. In: Workshop on Research Data Lifecycle Management, Princeton, NJ (July 2011)
5. Grzenda, M.: Towards the reduction of data used for the classification of network flows. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012, Part II. LNCS, vol. 7209, pp. 68–77. Springer, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-28931-6_7
6. Kim, H., Claffy, K., Fomenkov, M., Barman, D., Faloutsos, M., Lee, K.: Internet traffic classification demystified: myths, caveats, and the best practices. In: Proceedings of the 2008 ACM CoNEXT Conference, CoNEXT 2008, pp. 11:1–11:12. ACM, New York (2008)
7. Kobiersky, P., Korenek, J., Polcak, L.: Packet header analysis and field extraction for multigigabit networks, pp. 96–101 (April 2009)
8. Limwiatkul, L., Rungsawang, A.: Distributed denial of service detection using tcp/ip header and traffic measurement analysis, vol. 1, pp. 605–610 (October 2004)
9. Moore, A., Crogan, M., Moore, A.W., Mary, Q., Zuev, D., Zuev, D., Crogan, M.L.: Discriminators for use in flow-based classification. Tech. rep. (2005)
10. Ouyang, T., Ray, S., Rabinovich, M., Allman, M.: Can network characteristics detect spam effectively in a stand-alone enterprise? In: Spring, N., Riley, G.F. (eds.) PAM 2011. LNCS, vol. 6579, pp. 92–101. Springer, Heidelberg (2011)
11. Schatzmann, D., Burkhart, M., Spyropoulos, T.: Flow-level characteristics of spam and ham (291) (August 2008)
12. Žádník, M., Michlovský, Z.: Is spam visible in flow-level statistics? Tech. rep. (2009), http://www.fit.vutbr.cz/research/view_pub.php?id=9277

Monitoring Mental Fatigue through the Analysis of Keyboard and Mouse Interaction Patterns

André Pimenta, Davide Carneiro, Paulo Novais, and José Neves

CCTC/DI - Universidade do Minho
Braga, Portugal
pg20189@alunos.uminho.pt,
{dcarneiro,pjon,jneves}@di.uminho.pt

Abstract. In our living, we often have a sense of being tired due to a mental or physical work, plus a feeling of performance degradation even in the accomplishment of simple tasks. However, these mental states are often not consciously felt or are ignored, an attitude that may result in human failures, errors and even in the occurrence of health problems or on a decrease in the quality of life. States of fatigue may be detected with a close monitoring of some indicators, such as productivity, performance or even the health states. In this work it is proposed a model and a prototype to detect and monitor fatigue based on some of these items. We focus specifically on mental fatigue, a key factor in an individual's performance. With this approach we aim to develop leisure and work context-aware environments that may improve the quality of life and the individual performance of any human being.

Keywords: Mental Fatigue, Monitoring, Fatigue Detection, Behavioral Biometric.

1 Introduction

Fatigue is regarded as one of the main causes of human error. Many times its symptoms are ignored, as well as its importance for a good mental and physical condition, elementary for human performance and health [1,2]. The activity of driving a vehicle is a good example of the importance of fatigue in our activities or tasks, in which small errors can often and easily have a significant impact on human lives. This is even more significant in high-risk jobs such as aviation, transportation, aerospace, military or medicine, in which individuals routinely operate complex systems with a high degree of responsibility [2,3]. The study of this topic is thus decisive as the continuous disregard of the effects of fatigue, seen frequently as normal consequences of our busy lifestyle, may end up affecting one's quality of life, health or even lost of life.

Fatigue is however a very subjective concept and difficult to define from a scientific point of view. It may be seen as a combination of symptoms that include loss in performance (e.g. attention loss, slow reaction to a particular event, or

low performance in tasks to which the individual has the necessary skills), and subjective feelings of drowsiness and tiredness. From an abstract point of view, fatigue may be seen as two-dimensional, namely mental and physical. Despite the frequent inter-dependence of these two dimensions, they may be addressed independently [2].

In this work it is detailed a monitoring system for mental fatigue. It intends to detect different mental fatigue states of an individual in a non-invasive and transparent way. The system will analyze a set of features that stem from the individual's use of the computer, namely from the mouse and the keyboard. The system goes through a prior learning phase in which the behavior of an individual while using the computer without fatigue is studied. Later, it classifies its level of mental fatigue by quantifying changes in the individual's behavior. This approach will open the door to the development of better and intelligent working environments, that may be aware of their users' mental states and take actions towards the improvement of quality of life and their performance on tasks in which they are engaged.

2 Mental Fatigue Detection

Fatigue is a non-specific symptom. It may be estimated or detected from multiple sources, including the profile of the individual (e.g. age, gender, professional occupation, consumption of alcohol or drugs), performance and precision indicators (e.g. mouse click/movement precision or tasks delivered), or attention span (e.g. time spent on a particular task versus the time spent in other non-related tasks) [4].

The user's profile provides valuable information with respect to the potential level of fatigue [5]. It can be seen as a predicted base level of fatigue in the sense that it establishes a baseline, according to the lifestyle of the individual. These aspects have been thoroughly studied, mostly by psychologists, and encompass:

- **Age** - Defines de mental age of the individual. It is crucial to understand the expected cognitive abilities of the individual, which may have a tendency to degrade over time.
- **Gender** - The mental states are different between men and women.
- **Professional occupation** - Important to understand possible causes of mental fatigue, or to foil false states, since many occupations are intrinsically more tiresome or exhausting than others.
- **Consumption of alcohol and drugs** - The use of certain substances for short or prolonged periods of time may cause dependencies and other effects that lead to a state of mental fatigue.

Mental performance is generally seen as the combination of the speed and success/accuracy of an individual when carrying out a task. It is improved when the individual has plain use of his cognitive skills and decreased when these skills are somehow impaired or diminished [6]. Cognitive skills may include:

- **Memory** - The use of our memory includes all the processes related to encoding, storage and retrieval of information in our brain. Memory loss and loss of performance in accessing memory is clearly linked to mental fatigue, besides the factor of deteriorating with age.
- **Reaction time** - The reaction time is the timespan between the occurrence of a stimulus and the response of our body or mind. Responses may vary from a thought or a change in the mental state to a physical movement or an alteration in our physiology. Independently of the nature of the stimulus and the response, slow reaction times are usually associated with mental fatigue.
- **Concentration** - Concentration is the cognitive process of selectively focusing on one aspect of the environment while ignoring the others. The loss of concentration may be caused by external factors and may sometimes even be desirable (e.g. a sudden potentially hazardous event that must be analyzed before continuing with a given task). However, we frequently lose interest or focus in our tasks, at a rate that increases with the increase of the mental fatigue.
- **Accuracy and precision** - Accuracy and precision may be seen as the achievement of results that are within the expected quality and timeframe for a given individual, given his cognitive and physical skills. It also represents the absence of unexpected errors. This is among the factors that are more easily observable and measured while, at the same time, closely related to fatigue.

Mental fatigue is also affected by a number of other external factors. They may or may not be directly related to the individual's sphere, thus adding to the complexity of their comprehension and study [5,7]:

- **Mood** - The mood of the individual may influence decisively his/her the mental state, with a particular effect on his/her motivation to work. Although tired, the individual may overcome (even if only temporarily) the effects of fatigue with a positive mood and motivation.
- **Stress** - Stress may be defined as the set of responses of the individual's mind or body to external stimuli, allowing the individual to adapt to the dynamic requirements of the environment. These processes of adaptation, however, require an additional effort from the brain which, when prolonged over long or intense periods of time, will result in mental fatigue.
- **Sleepiness** - Sleepiness is often mistaken for mental fatigue or generalized as such. The difference exists and must be pointed out. However, the mistake is understandable since sleepiness is a symptom that is strongly connected to mental fatigue: it is one of the ways our brain uses to tell us that he is running out of resources. It often results in a general loss of the individual vitality.

2.1 Indicators of Mental Fatigue

The study of mental fatigue, including its causes and symptoms, is traditionally supported by self-reporting mechanisms (generally questionnaires) or, more recently, through the use of physiological sensors. The first approach, related to

Psychology, has a certain number of disadvantages: (1) people often lie or hide truths in questionnaires; (2) people are afraid or unwilling to answer correctly; (3) people frequently have wrong and subjective perceptions of their symptoms; (4) questionnaires are often hard to define correctly, with inaccurate questions or answers. The second approach, followed by Medicine, has the advantage of being more accurate. However, this has its price on the invasive and intrusive nature of the physiological sensors used, which may even influence the variables of the study[8,9].

In order to study fatigue we are thus following an approach already validated in the study of stress [10,11]: analyzing changes in the behavior of the user while within a given technologically empowered environment, in a non-intrusive and non-invasive way. This results in an environment that may adjust, in real-time, to significant changes in its context, with context being defined by the level of fatigue of the individuals.

The detection and classification of fatigue will be based on the collection of data about behavioral biometrics, specifically keystroke and mouse dynamics. A simple logger application was developed that acquires information about each mouse and keyboard action (e.g. mouse button down, mouse button up, key down, key up, mouse movement, among others), with the detail needed to generate the features pointed out below. Following this approach it is possible to collect data that will allow to discover behavioral patterns of interaction with the keyboard and mouse, in a non-intrusive and dynamic way. Particularly, the following features are considered:

- **Keydown Time** - time spent between the key down and the key up events;
- **Errors per Key Pressed** - number of times the backspace key is pressed, versus the keys pressed;
- **Mouse Velocity** - velocity at which the cursor travels;
- **Mouse Acceleration** - acceleration of the mouse at a given time;
- **Time between Keys** - time spent between each two keys pressed;
- **Total Excess of Distance** - excess of distance travelled by the pointer when considering two consecutive clicks;
- **Average Excess of Distance** - average of the distance excess travelled by the pointer when considering two consecutive clicks;
- **Double Click Speed** - speed of the double click;
- **Number of Double Clicks** - number of double clicks in a time frame;
- **Distance While Clicking** - distance travelled by the mouse while dragging objects;
- **Signed Sum of Angles** - how much the pointer "turned" left or right during its travel;
- **Absolute Sum of Angles** - how much the pointer "turned" during its travel, in absolute terms;
- **Sum of Distances between Path and Straight Line** - considering two consecutive clicks, it measures the distance between all the points of the path travelled by the mouse, and the closest point in a straight line (that represents the shortest path) between the coordinates of the two clicks;

- **Average Distance between Path and Straight Line** - the same as above, but provides an average value of the distance to the straight line; and
- **Time between Clicks** - time spent between each two clicks.

3 The Mental Fatigue Monitoring Framework

As seen in section 2, indicators of mental fatigue are defined by a set of metrics. In order to have a positive and optimized effect on the lives of its users, a framework for fatigue monitoring based on these metrics must perform classification and decision-making in real-time.

The architecture of the proposed framework includes not only the sheer acquisition and classification of the data, but also a presentation tier that will support the human-based or autonomous decision-making mechanisms that are now being implemented. It is a layered architecture. The first layer is the *Data acquisition* one. It is responsible for capturing information describing the behavioral patterns of the user, receiving data from events fired from the use of the mouse and the keyboard. Therefore, this layer encodes each event with the corresponding necessary information (e.g. timestamp, coordinates, type of click, key pressed).

The second layer is the *Data processing*. In this layer the data received from the *Data acquisition* layer is processed and transformed in order to be evaluated according to the metrics presented. One of the most important tasks of this layer is to filter outlier values that would have a negative effect on the analysis (e.g. a key pressed for more than a certain amount of time).

The next layer is the *Classification layer*. This layer is responsible for interpreting data from the mental fatigue indicators and to build the meta data that will support decision-making. To do it, this layer uses the machine learning mechanisms detailed below.

After the classification, the *Data access layer*, is responsible for providing access to - and persisting all the information generated by - the lower layer. We are not only interested in allowing access to the data in real-time but also to persist the historic of the user as well as his/her profile, to allow analyses within longer time frames.

Finally, the *Presentation layer* includes the mechanisms to build intuitive and visual representations of the mental states of the users, abstracting from the complexity of the data level where they are positioned.

3.1 Classifying Fatigue

The classification of the mental state of a user is achieved through the use of the k-Nearest Neighbor algorithm (k-NN). It is a method of classification based on closest training examples in the feature space. The data used to train the model used by the k-NN algorithm was collected in an experiment performed with students of the University of Minho, as described in section 4. From the available features, only the ones showing the most statistical significance were

selected: Mouse Acceleration, Mouse Velocity, Keydown Time, Time between keys, and Error per key.

A different classifier is trained for each feature in order to get the level of fatigue due to each one. This is of significant importance since there are periods of time during which the framework only accesses a limited set of features (e.g. the user may not use the keyboard for a certain period of time). Thus, each feature is classified independently, assigning a binary value to each of its inputs, marking it as fatigued or not fatigued.

Afterwards, a weighted sum is computed involving all the features with available data. The weights of the features for the overall computation of the level of fatigue are given by the significance of the feature, computed during the statistical analysis of the data at hand, as described in section 4. This means that features that have been significantly affected by fatigue will have a larger contribution to the computed value than the ones that were hardly affected during the experiment. This weighted sum constitutes a fusion of the different features, resulting in the actual level of fatigue for a user. Using training data for a specific individual, when available, makes the fatigue detection model personalized, thus more accurate.

The prototype of the presentation layer (Figure 1) uses a set of emoticons to depict the state of each user in an intuitive way that is easy to understand. For more detail it is also possible to access a more detailed interface, which shows not only the overall state of fatigue of the user (in a binary form as fatigued or not fatigued) but also the level of fatigue, which actually represents the weighted sum of the features.

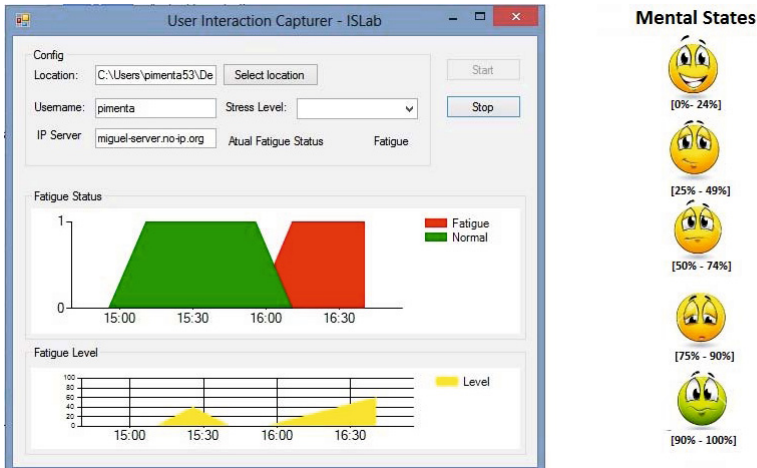


Fig. 1. The presentation layer of the monitoring system detailing the level of fatigue of an individual in part of the afternoon

4 Analysis of Results

Twenty participants took part on the experiment in order to collect data to train the fatigue detection module. In this experiment participants needed only have an application running that recorded all their interactions with the keyboard and the mouse. It did not interfere with their usage patterns nor needed the participants to perform additional or different activities. The participants (seventeen males and three females) were mostly volunteer students from our institution, aged eighteen to fifty. All these individuals were familiar with the technological devices used and the interaction with them was not an obstacle.

The collection of the data took place in two moments, for each user: the first at the beginning of the day, where participants are fully rested and in a normal mental state and the second at the end of the day. The two collection moments took place on the same day. It is also important to mention that the collection of data took place during the exam season and that the participants spent the day studying, with the support of a computer.

4.1 Statistical Data Analysis

To determine to which extent each feature considered is or is not influenced by mental fatigue, the data collected in the first phase (without fatigue) is compared with the data collected in the second phase (with expected mental fatigue). Given that most of the distributions of the data collected are not normal, the Mann-Whitney test is used to perform the analysis. This test is a non-parametric statistical hypothesis test for assessing whether one of two samples of independent observations tends to have larger values than the other. The null hypothesis is thus: $H_0 = \text{the medians of the two distributions are equal}$. For each two distributions compared, the test returns a p -value, with a small p -value suggesting that it is unlikely that H_0 is true. Thus, for every Mann-Whitney test whose p -value $< \alpha$, the difference is considered to be statistically significant, i.e., H_0 is rejected, with $\alpha = 0.05$. The data analysis was performed using Wolfram Mathematica, Version 8.0.

A significant difference between the collected data in the two phases means that the feature is effectively influenced by mental fatigue for this specific individual. Statistically significant differences were observed in the features Keydown Time, Errors per Key Pressed, Time Between Keys, Mouse Velocity and Mouse Acceleration, hence only these are used for building the model.

For each feature and for each of the two moments of data collection, the average and median values were analyzed in order to determine the trend of the value, i.e., we wanted to answer the question "does it tend to decrease or to increase under fatigue?". Table 1 summarizes these observations.

The results obtained lead us to conclude that all the features have a significantly marked trend, either increasing or decreasing. The only exception to this is the feature **Time between Clicks**, which increased to half of the participants and decreased to the other half.

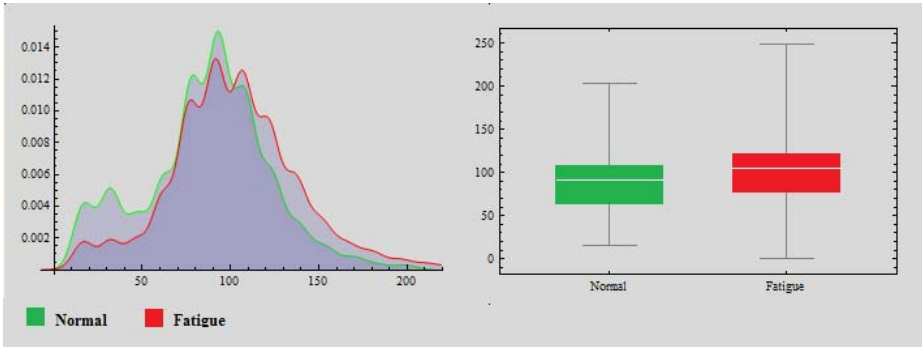


Fig. 2. Histograms and Box Plots comparing the data of the two distributions for the feature **Keydown Time** of a specific volunteer: fatigued individuals tend to write slower

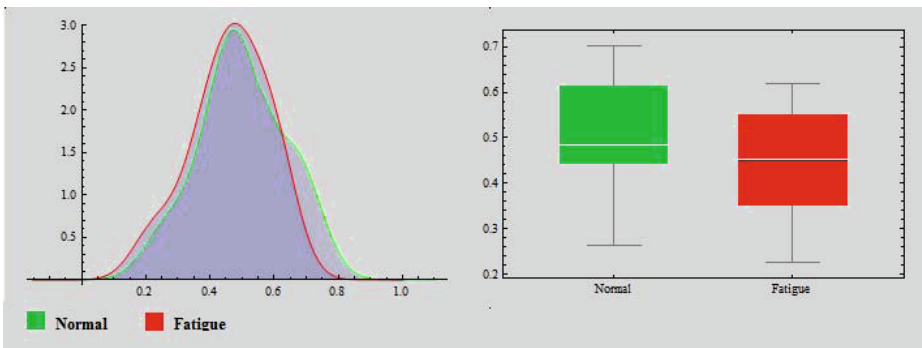


Fig. 3. Histograms and Box Plots comparing the distributions of the data collected in the two moments for the feature **Mouse Velocity** of a volunteer: fatigued individuals move the mouse slower.

After the statistical analysis performed, one may claim that it is indeed possible to detect mental fatigue through the handling of keyboard and mouse usage data. It also proves, once more, that the presence of mental fatigue is accompanied by a loss of performance and an increase in errors, and that this loss may be measured using non-invasive tools. This loss of performance and increase in the number of errors are consequences of decreased cognitive skills, which are caused by the increasing level of fatigue being measured during the day's work. The decrease in the cognitive skills is evidenced by general patterns of slower mouse (mouse acceleration and velocity) and keyboard (Keydown Time, Time between keys) interactions, as well as by an increasing number of errors (Error per key).

Table 1. Results of the statistical analysis of the data for the 20 participants. Only the features that have shown significant differences are included. The "trend" column depicts the percentage of participants that have a given trend. For example, the mean value of the velocity of the mouse decreases for 90% of the students, when fatigued

Metric		Normal	Fatigued	Trend	p-value
Keydown Time	Mean:	79.827	87. 119	Increases in 100%	$0.7 * 10^{-4}$
	Median:	77.601	81.502	Increases in 60%	
Time between keys	Mean:	469. 193	1040.26	Increases in 100%	$1.23 * 10^{-144}$
	Median:	215.75	386.55	Increases in 90%	
Mouse Acceleration	Mean:	0.4238	0.3829	Decreases in 90%	$3.01 * 10^{-11}$
	Median:	0.2202	0.2010	Decreases in 100%	
Mouse Velocity	Mean:	0.5002	0.4401	Decreases in 90%	$5.03 * 10^{-15}$
	Median:	0.2680	0.2537	Decreases in 100%	
Time between Clicks	Mean:	3081.35	3257.61	Increases in 50%	$5.8 * 10^{-4}$
	Median:	1733.30	1863. 15	Increases in 50%	
Error per key	Mean:	7.643	9.002	Increases in 90%	$2 * 10^{-2}$
	Median:	7.444	8.598	Increases in 90%	

5 Conclusion

In this paper it is presented an approach to classify the level of mental fatigue of the individuals using a computer, by analyzing their interaction patterns, specifically the aspects related to the use of the mouse and the keyboard. The most noteworthy aspects of the work presented is that it details a non-invasive, non-intrusive and transparent approach to solve the problem. Existing approaches are either based on questionnaires or on physiological sensors, both with disadvantages of their own. The approach presented is based on the analysis of the user's behaviour.

The results obtained prove not only the effect of fatigue on the user's performance throughout the day but also that it is possible to measure and classify these effects, in real time. However, some issues are still open. Specifically we are now working on the identification of the user from the interaction patterns. This will allow to determine if the data being used to classify fatigue actually belongs to the expected user or if the data or parts of it may belong to another user that might have been using the computer. This input will be used to improve the accuracy of the classification.

This work opens the door to the development of leisure and working environments that are aware of their user's level of fatigue, and may, therefore, provide decision-support systems that will improve their working performance and their quality of life.

Acknowledgements. This work is part-funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science

and Technology) within project FCOMP-01-0124-FEDER-028980 (PTDC/EEI-SII/1386/2012) and PEst-OE/EEI/UI0752/2011. The work of Davide Carneiro is also supported by a doctoral grant by FCT (SFRH/BD/64890/2009).

References

1. van der Linden, D., Eling, P.: Mental fatigue disturbs local processing more than global processing. *Psychological Research* 70(5), 395–402 (2006)
2. Williamson, R.J., Purcell, S., Sterne, A., Wessely, S., Hotopf, M., Farmer, A., Sham, P.C.: The relationship of fatigue to mental and physical health in a community sample. *Social Psychiatry and Psychiatric Epidemiology* 40(2), 126–132 (2005)
3. Nussbaum, M.A., Babski-Reeves, K.L., Kleiner, B.M., Smith-Jackson, T.L., Barker, L.M.: Measuring and modeling the effects of fatigue on performance: Specific application to the nursing profession Linsey Marinn Barker Virginia Polytechnic Institute and State University to partially fulfill the requirements of a Doctor of Philosophy In Indus. PhD thesis (2009)
4. Winwood, P.C., Winefield, A.H., Dawson, D., Lushington, K.: Development and Validation of a Scale to Measure Work-Related Fatigue and Recovery: The Occupational Fatigue Exhaustion/Recovery Scale (OFER). *Journal of Occupational and Environmental Medicine* 47(6), 594–606 (2005)
5. Kobayashi, H., Demura, S.: Relationships between Chronic Fatigue, Subjective Symptoms of Fatigue, Life Stressors and Lifestyle in Japanese High School Students. *School Health* 2, 5 (2006)
6. Joyce, E., Blumenthal, S., Wessely, S.: Memory, attention, and executive function in chronic fatigue syndrome. *Journal of Neurology, Neurosurgery, and Psychiatry* 60(5), 495–503 (1996)
7. Hossain, J.L., Ahmad, P., Reinish, L.W., Kayumov, L., Hossain, N.K., Shapiro, C.M.: Subjective fatigue and subjective sleepiness: two independent consequences of sleep disorders? *Journal of Sleep Research* 14(3), 245–253 (2005)
8. Joyce, J., Rabe-Hesketh, S., Wessely, S.: Reviewing the reviews: the example of chronic fatigue syndrome. *JAMA: The Journal of the American Medical Association* 280(3), 264–266 (1998)
9. Neuberger, G.B.: Measures of fatigue: The Fatigue Questionnaire, Fatigue Severity Scale, Multidimensional Assessment of Fatigue Scale, and Short Form-36 Vitality (Energy/Fatigue) Subscale of the Short Form Health Survey. *Arthritis & Rheumatism* 49(S5), S175–S183 (2003)
10. Novais, P., Carneiro, D., Gomes, M., Neves, J.: Non-invasive Estimation of Stress in Conflict Resolution Environments. In: Demazeau, Y., Müller, J.P., Rodríguez, J.M.C., Pérez, J.B. (eds.) *Advances on PAAMS*. AISC, vol. 155, pp. 153–160. Springer, Heidelberg (2012)
11. Carneiro, D., Castillo, J.C., Novais, P., Fernández-Caballero, A., Neves, J.: Multi-modal Behavioural Analysis for Non-invasive Stress Detection (July 2012)

On Mining Sensitive Rules to Identify Privacy Threats

Irene Díaz¹, Luis J. Rodríguez-Muñiz², and Luigi Troiano³

¹ Computer Science Department, University of Oviedo, Spain
sirene@uniovi.es

² Department of Statistics and O.R., University of Oviedo, Spain
luisj@uniovi.es

³ Department of Engineering, University of Sannio, Italy
troiano@unisannio.it

Abstract. Data mining techniques represent a useful tool to cope with privacy problems. In this work an association rule mining algorithm adapted to the privacy context is developed. The algorithm produces association rules with a certain structure (the premise set is a subset of the public features of a released table while the consequent is the feature to protect). These rules are then used to reveal and explain relationships from data affected by some kind of anonymization process and thus, to detect threats.

Keywords: disclosure control, association rules, data privacy, anonymity.

1 Introduction

The huge increase in digital data has led to emerge some concerns about data privacy, especially nowadays because WWW makes easy linking information about individuals obtained from different sources. It is easy to access to micro-data (data that are not summarized by some statistics) which are generally organized in tables whose attributes can (i) lead to identities, such as address, name, social security number, and (ii) release sensitive information, such as diseases and income, such those regarding census, medical issues, finance and others. In particular those attributes that are directly linked to identity are know as *identifiers*, whilst other attributes related at some extent to identity potentially able to identify an individual are known as *quasi-identifiers*.

In this environment data mining techniques are becoming extremely important to infer hidden information from data collections by providing patterns or existing models in data collections. The work presented in this paper tries to take advantage of association rules to study the possible threats in a given anonymous table. Firstly, an association rule based algorithm is developed to obtain a rule base where the antecedent set is a subset (possibly the whole set) of the quasi-identifier set while the consequent of each rule is the sensitive variable. After that, we will study the robustness of an anonymous table against attacks

performed via this rule base. We will also study the conditions a rule must satisfy to be helpful in revealing information in a data set. It is worth to note that, differently from l -diversity and k -anonymity which are aimed at describing overall properties of released table, the proposed algorithm is aimed at describing single relationships between data, and thus to reveal leaks in the anonymization scheme.

The rest of the paper is organised as follows. Section 2 presents basic concepts in Data Privacy as well as the main metrics used to obtain an anonymous table. Section 3 shows some existing works in the context of data mining in privacy. Section 4 describes the RuleMiner algorithm. Finally, Section 5 shows an initial example of the procedure and Section 6 draws the conclusions and our future plans.

2 Privacy by Means of Metrics

Different metrics for measuring the level of privacy guaranteed by Statistical Disclosure Control (SDC) have been proposed over the time. They study the need for protection centred on limiting the ability to link released information to other external data. That limitation is controlled by identifying all attributes in the private information that could be used for linking with external information to uniquely identify individuals in the population. Such attribute set is named *quasi-identifier*.

Among them, in [15] and [17] is defined k -anonymity with respect to a quasi-identifier as the property that makes each record of a released table indistinguishable with at least $k - 1$ other records. Therefore, k -anonymity requires that each equivalence class contains at least k records. This property assures that if the released data satisfies k -anonymity with respect to a quasi-identifier, then the combination of the released data and the external sources on which the quasi-identifier is based, cannot link on the quasi-identifier or a subset of its attributes to match fewer than k records. Machanavajjhala *et al.* propose l -diversity [14] for providing privacy when the data publisher does not know what kind of information manages the attacker. Thus, l -diversity requires that the distribution of a sensitive attribute in each equivalence class has at least l values.

However l -diversity is limited in its assumption of adversarial knowledge. To avoid this lack, t -closeness [11] formalizes the idea of global background knowledge by requiring that the distribution of a sensitive attribute in any equivalence class is close to the distribution of the attribute in the overall table. This effectively limits the amount of individual-specific information an observer can learn.

Other approaches try to prevent range disclosure, i.e., when an attacker is able to link an individual not to a specific sensitive value, but to a set of values that collectively disclose some sensible information [13]. In this sense, fuzzy set theory provides a natural framework to analyze data generalization and to identify threats to privacy (see [5,4]). Since generalization is about grouping elements in classes, and membership cannot be sharply defined, a class of elements can be regarded as fuzzy set. Privacy is preserved and disclosure protected, if the

anonymization scheme chosen is able to mix sensitive data in such a way to make them indistinguishable at different level of generalization. Other existing approaches to prevent range disclosure are explained in [12] or [10].

Previous measures and some other more as p -sensitivity [16] drive data anonymization with respect to same aspect, but all of them share the common idea that having more records within a group associable to an entity enforce privacy protection.

3 Privacy as a Data Mining Problem

The relationship between data mining (DM) and statistical disclosure control (SDC) has been firstly outlined in [6]. The problem in attribute disclosure is basically to find an inferential path from released attributes to sensitive information, revealing personal information about individuals to whom information refers. Therefore, privacy protection in data mining is becoming a bottleneck nowadays.

DM [8] searches for the relationships that exist in large databases, but hidden due to the large amount of data. DM models the behavior of a given variable in terms of the others, finding non-trivial relationships among the attributes involved [9]. These relationships may provide valuable knowledge regarding the individuals the data are related to. As rule mining is aimed at reconstructing the hidden linkage of given patterns between attributes, it is able to put into evidence that if some knowledge is discovered by intruders, this can be used to break privacy protections. In this sense, DM may infer relationships that can lead to sensitive information disclosure.

The problem of mining association rules have been widely investigated in literature, and several search algorithms have been proposed. Among them, the most prominent is Apriori [7]. This algorithm and its variants perform an exhaustive search of rules with high support and confidence. Association rules algorithm have been used to discover threats in some previous works as [2,3].

If in data mining it is enough to infer models from training data sets in order to overcome data distortion, in SDC we are interested in models able to reveal and explain relationships in presence of data distortion, as generally introduced by the anonymization process. Indeed some information, although hidden or even removed, could be still linked to identities at some extent considering a lower level of data granularity, at which different point-wise information are assimilated. This problem has been raised in [11].

4 SRM Algorithm

As it can be seen in Section 2, to limit the risk of releasing sensitive information to an acceptable level the data are made anonymous by removing explicit references to identity and by replacing the other attribute values related to identity with values less specific but semantically consistent. This leads to group records with the same quasi-identifier values into *equivalence classes*. In an equivalence class,

individuals are supposed to be indistinguishable. But the equivalence classes could leak information due to lack of diversity in the sensitive attribute. Even more, sometimes the anonymous released table is not protected against attacks based on background knowledge [14]. In this section we introduce the concept of *sensitive rules*, as special case of the more general association rules, and we outline the Sensitive Rule Miner (SRM) algorithm as an efficient means to discover them.

4.1 Basic Concepts

Let \mathcal{D} be a dataset whose schema is made by attributes A_1, \dots, A_n . An association rule describes co-occurrence of values. Formally speaking, an association rule is defined as a relation of the form $X \longrightarrow Y$, where X and Y are subsets of pairs $A_i = v_{ij}$ with v_{ij} is one of the possible values assumed by the attributed A_i . We assume $X \cap Y = \emptyset$.

The support (*supp*) of $X \longrightarrow Y$ represents the fraction of tuples in \mathcal{D} assumes values expressed by X , while the confidence (*conf*) measures how often values expressed by Y co-occur with values expressed by X . Formally,

$$\text{conf}(X \longrightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)} \quad (1)$$

In other terms, the support provides a measure of how often a combination of values is presented, while confidence how often the association between values occur.

The problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum support (called *minsup*) and minimum confidence (called *minconf*) respectively. The most standard algorithm to cope with this problem is Apriori. For a detailed description, authors refer to [1].

By contrast, in the context of Privacy Preserving in a database, we look for rules to discover threats and thus, a rule covering just one transaction could be important if the information about one individual is revealed. Therefore, a straightforward application of any rule mining algorithm is not useful to cope with the problem of studying disclosure control problem.

This work presents an association rule algorithm supported by Apriori algorithm but adapted to the privacy preserving problem. The main differences with regard to other existing approaches are listed below. The algorithm looks for rules with

- a high confidence, but not necessarily a high support, because any possible thread is interesting to discover, even if it allows us to discover just one transaction.
- a certain structure. Any association rule algorithm (for example Apriori) looks for association rules among the variables in large databases. However it is not relevant the variable distribution along the antecedent and the consequent. By contrast, we are interested to discover associations between

quasi-identifier values P to sensitive information Q . We call rules with this structure $P \rightarrow Q$, *sensitive rules*.

- a minimal structure. Given two rules $P_1 \rightarrow Q$ and $P_2 \rightarrow Q$, if $P_1 \subset P_2$, the rule $P_1 \rightarrow Q$ is preferred.

As a consequence, if $T = \{t_1, t_2, \dots, t_m\}$ is the set of features, $P = \{t_{\sigma(1)}, \dots, t_{\sigma(n)}\}$ with $n < m$ and $\sigma()$ a permutation function is the quasi-identifier, instead of exploring the power set of T , we explore the power set of P as the consequent of the rules is fixed.

Suppose we have an anonymized data set whose quasiidentifier is composed by three variables, X_1, X_2, X_3 . Suppose that the possible values for X_1 are a and b , r, s, t for X_2 and u for X_3 . Figure 1 shows the lattice of the quasi-identifier, i.e., the lattice of the premises.

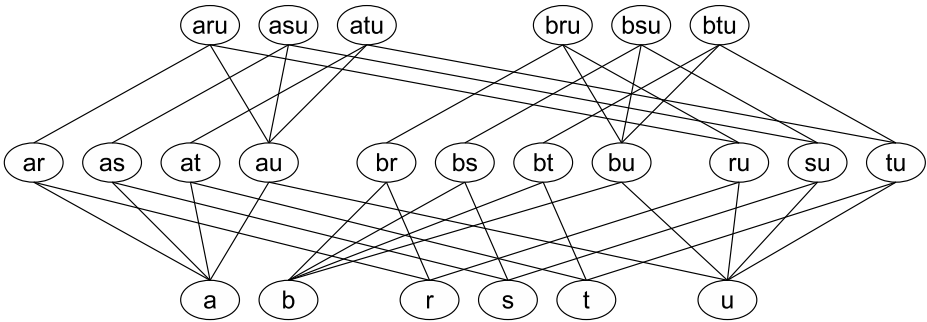


Fig. 1. Lattice of possible rule premises

4.2 The Algorithm

The algorithm presented here mines sensitive rules over a given confidence and support traversing level-wise the premise lattice from the bottom to the top (i.e., starting from the shortest itemset to the largest).

Algorithm 1 shows the procedure to obtain the sensitive rules. Given a lower bound for the support (t_s) of the premise set and the confidence of an association rule (t_c), the algorithm searches for rules with a minimal structure. To do that, for any combination of sensitive values (Q), the confidence (c_U in Algorithm 1) of any rule $P \rightarrow Q$ with $|P| = 1$ is computed. If c_U is over t_c and the part of the lattice containing the premise P is not further studied (because we look for minimal rules).

This process is then iterated over the unseen premise sets with cardinality 2. The process halts when the set of possible premises is empty. Note that the candidate set of premises with cardinality n is dynamically built from the premise sets with cardinality $n - 1$ and also depends on the k -anonymity and l -diversity of the table. In fact, note that given any rule $P \rightarrow Q$, its confidence is bounded by k and l as follows

$$\text{conf}(P \rightarrow Q) = \frac{\text{supp}(P \cup Q)}{\text{supp}(P)} \leq \frac{\text{supp}(P \cup Q)}{k} \quad (2)$$

because by definition of k -anonymity if P represents the whole quasiidentifier, $\text{supp}(P) = k$, otherwise $\text{supp}(P) \geq k$. Besides, the number of cooccurrences of any pair P, Q in a k -anonymous, l -diverse table is at most $k - l + 1$, therefore

$$\text{conf}(P \rightarrow Q) = \frac{\text{supp}(P \cup Q)}{\text{supp}(P)} \leq \frac{k - l + 1}{\text{supp}(P)} \quad (3)$$

Since $\text{supp}(P) \geq k \forall P$, from Eq.(2) and Eq.(3), we get

$$\text{conf}(P \rightarrow Q) = \frac{\text{supp}(P \cup Q)}{\text{supp}(P)} \leq \frac{k - l + 1}{k} \quad (4)$$

Note that if some feature in the data set is numeric, a discretization of values will be required before starting the process, because SRM algorithm only works with nominal attributes, but this is out of our scope in this work.

Let's mine the lattice of premises shown in Figure 1 according to the Algorithm 1. Figure 2 shows the thresholds for confidence (t_c) and support (t_s). Note that the confidence is not a monotonic function, but according to absorption rule it is possible to find a convex region. On the other side, the support of sets over t_s line in Figure 2 is below the threshold t_s . Therefore we are interested in mining rules whose premises are within t_s and t_c lines.

Suppose the sensitive value is q . As the algorithm follows a bottom-up approach, the first premises it studies are $\{a, b, r, s, t, u\}$. As $\text{supp}(a) \geq t_s$, the rule $a \rightarrow q$ is considered.

As $\text{conf}(a \rightarrow q) \geq t_c$, $a \rightarrow q$ is proposed as association rule and any rule $P \rightarrow q$ with $a \in P$ is discarded. Now $\{b\}$ is studied. Again as $\text{supp}(b) \geq t_s$, the rule $b \rightarrow q$ is also considered. As $\text{conf}(b \rightarrow q) \geq t_c$, $b \rightarrow q$ is proposed as association rule and any rule $P \rightarrow q$ with $b \in P$ is discarded.

This process is then repeated for all single premises. According to Figure 2, if the premise r is evaluated, as $\text{conf}(r \rightarrow q) < t_c$, the rule is not added to the rule base (even with $\text{supp}(r) \geq t_s$). As confidence is not monotonic, the algorithm explores the premises containing r but not a and b (because the rules $a \rightarrow q$ and $b \rightarrow q$ are already added to the rule base). Following this procedure, the complete rule base is obtained.

5 Running Example

Finally, we show with a small example, how the system identifies possible threats. Table tab:anonymity shows an example of a 3-anonymised data. The quasi-identifier is composed by the variables *ZIP Code* and *Age* while the sensitive value is *Disease*.

We studied the algorithm behaviour at different levels of confidence, ranging from 0 to 1 with 0.1 steps. When the algorithm looks for rules with low confidence

Algorithm 1. SRM Algorithm

Input: \mathcal{D} , dataset**Input:** t_s , minimum support**Input:** t_c , minimum confidence**Output:** \mathcal{R} , the set of rules $P \rightarrow Q$.

```

{Step 1: Init}
1:  $A_1, \dots, A_p \leftarrow$  quasi-identifier attributes of  $\mathcal{D}$ 
2:  $B_1, \dots, B_q \leftarrow$  sensitive attributes of  $\mathcal{D}$ 
3:  $\mathcal{U} \leftarrow$  all sensitive values in  $B_1, \dots, B_q$ 
4:  $\mathcal{V} \leftarrow$  all quasi-identifier values in  $A_1, \dots, A_p$ 
5:  $Z \leftarrow \emptyset$ , vetoed values in  $V$ 
6:  $k \leftarrow$  anonymity of  $\mathcal{D}$ 
7:  $l \leftarrow$  diversity of  $\mathcal{D}$ 
{Step 2: Search}
8:  $F \leftarrow V$ 
9:  $E \leftarrow \emptyset$ 
10:  $\mathcal{U}_P \leftarrow \mathcal{U} \ \forall P \in F$ 
11: repeat
12:   for all  $P \in F$  do
13:      $s \leftarrow$  support of  $P$ 
14:     if  $s \geq t_s$  then
15:       for all  $Q \in \mathcal{U}_P$  do
16:          $c_U \leftarrow$  confidence of  $P \rightarrow Q, Q \in \mathcal{U}_P$ 
17:         if  $c_U \geq t_c$  then
18:           add  $P \rightarrow Q$  to  $\mathcal{R}$ 
19:            $Z \leftarrow Z \cup P$ 
20:         end if
21:       end for
22:        $\mathcal{U}'_P \leftarrow \{Q \in \mathcal{U} | c_u < t_c\}$ 
23:     else
24:        $\mathcal{U}'_P \leftarrow \mathcal{U}_P$ 
25:     end if
26:     if  $\mathcal{U}'_P \neq \emptyset$  and  $\min(\frac{k-l+1}{\text{supp}(P)}, \frac{\text{supp}(PQ)}{k}) \geq t_c \ \exists Q \in \mathcal{U}'_P$  then
27:        $\mathcal{S}_P \leftarrow \{S \supset P | S \cap Z = \emptyset\}$ 
28:        $\mathcal{U}_S = \mathcal{U}'_P \ \forall S \in \mathcal{S}_P$ 
29:        $E \leftarrow E \cup \mathcal{S}_P$ 
30:     end if
31:   end for
32:    $F \leftarrow E$ 
33:    $E \leftarrow \emptyset$ 
34: until  $F$  is empty

```

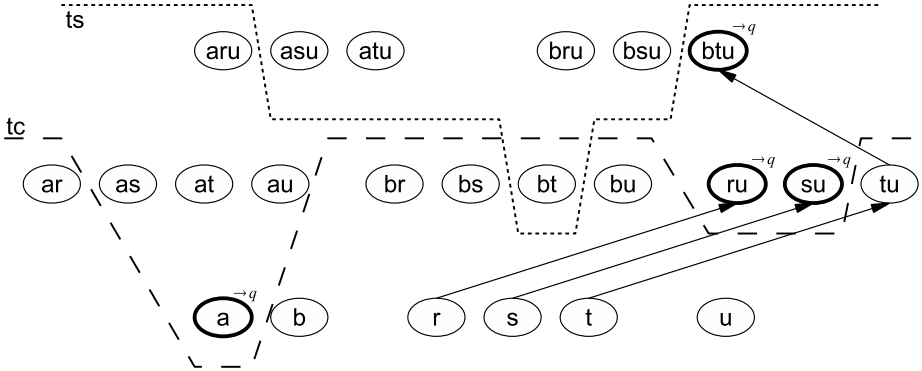


Fig. 2. Example of algorithm applied to consequence q

Table 1. Anonymization induced by 3-anonymity

	ZIP Code	Age	Salary	Disease
1	476**	2*	3K	gastric ulcer
2	476**	2*	4K	gastritis
3	476**	2*	5K	stomach cancer
4	479**	[40, 55]	6K	gastritis
5	479**	[40, 55]	11K	flu
6	479**	[40, 55]	8K	bronchitis
7	476**	3*	7K	bronchitis
8	476**	3*	9K	pneumonia
9	476**	3*	10K	stomach cancer
10	479**	≥ 55	9K	heart attack
11	479**	≥ 55	9K	heart attack
12	479**	≥ 55	10K	angina pectoris

(0.2), it finds many rules basically describing the information contained in the released table.

- IF ZIP CODE(476**) THEN DISEASE (stomach cancer)*
- IF ZIP CODE(476**) THEN DISEASE (heart attack)*
- IF AGE(2*) THEN DISEASE (gastric ulcer OR gastritis OR stomach cancer)*
- IF AGE([40-55]) THEN DISEASE (gastritis OR flu OR bronchitis)*
- IF AGE(3*) THEN DISEASE (bronchitis OR pneumonia OR stomach cancer)*
- IF AGE(≥ 55) THEN DISEASE (heart attack OR angina pectoris)*

If the minimum confidence is increased (ranging from 0.3 to 0.6), we just obtain the rule *IF AGE(≥ 55) THEN DISEASE (heart attack)* and for confidences over 0.6 we don't obtain any rule. This is according to the fact that table has 2 – diversity and 3 – anonymity, so that upper bound to confidence is

$$\frac{k - l + 1}{k} = \frac{2}{3}$$

6 Conclusions and Future Work

This work proposes SRM, an association rule based algorithm to discover threats in anonymous data bases, by discovering those rules able to link quasi-identifiers to sensitive values. It is worth to note that, differently from l-diversity and k-anonymity which are aimed at describing overall properties of released table, SRM is aimed at describing single relationships between data, and thus to reveal leaks in the anonymization scheme. So, it becomes complementary to usual approaches based on metrics. Among the future challenges, we plan to compare these results to the obtained when an anonymous table is attacked by the rule base provided by some well known classification methods. In addition, in a near future, the algorithm will be adapted to work with continuous variables.

Acknowledgements. Authors acknowledge financial support Grants MTM2011-22993 and TEC2012-38142-C04-04 from Ministry of Education and Science, Government of Spain.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB 1994, pp. 487–499. Morgan Kaufmann Publishers Inc., San Francisco (1994)
2. Atzori, M., Bonchi, F.: Blocking anonymity threats raised by frequent itemset mining. In: ICDM, pp. 561–564. IEEE Computer Society (2005)
3. Ciriani, V., Vimercati, S., Foresti, S., Samarati, P.: k-anonymous data mining: A survey. In: Aggarwal, C., Yu, P. (eds.) Privacy-Preserving Data Mining. Advances in Database Systems, vol. 34, pp. 105–136. Springer US (2008)
4. Díaz, I., Ranilla, J., Rodríguez-Muniz, L.J., Troiano, L.: Identifying the risk of attribute disclosure by mining fuzzy rules. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) IPMU 2010. CCIS, vol. 80, pp. 455–464. Springer, Heidelberg (2010)
5. Díaz, I., Rodríguez-Muñiz, L.J., Troiano, L.: Fuzzy sets in data protection: strategies and cardinalities. *Logic Journal of IGPL* 20(4), 657–666 (2012)
6. Domingo-Ferrer, J., Torra, V.: On the connections between statistical disclosure control for microdata and some artificial intelligence tools. *Inf. Sci. Inf. Comput. Sci.* 151, 153–170 (2003)
7. Evfimievski, A.V., Srikant, R., Agrawal, R., Gehrke, J.: Privacy preserving mining of association rules. In: KDD, pp. 217–228. ACM (2002)
8. Holsheimer, M., Siebes, A.P.: Data mining: the search for knowledge in databases. Technical report, Amsterdam, The Netherlands (1994)
9. Laird, P.D.: Learning from good and bad data. Kluwer Academic Publishers, Norwell (1988)
10. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Workload-aware anonymization techniques for large-scale datasets. *ACM Trans. Database Syst.* 33(3), 17:1–17:47 (2008)
11. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: Chirkova, R., Dogac, A., Özsu, M.T., Sellis, T.K. (eds.) ICDE, pp. 106–115. IEEE (2007)

12. Loukides, G., Shao, J.: Capturing data usefulness and privacy protection in k-anonymisation. In: Proceedings of the 2007 ACM Symposium on Applied Computing, SAC 2007, pp. 370–374. ACM, New York (2007)
13. Loukides, G., Shao, J.: Preventing range disclosure in k-anonymised data. *Expert Syst. Appl.* 38(4), 4559–4574 (2011)
14. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data* 1(1) (March 2007)
15. Samarati, P.: Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering* 13, 1010–1027 (2001)
16. Sun, X., Wang, H., Li, J., Ross, D.: Achieving p-sensitive k-anonymity via anatomy. In: Proceedings of the 2009 IEEE International Conference on e-Business Engineering, pp. 199–205. IEEE Computer Society, Washington, DC (2009)
17. Sweeney, L.: k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* 10(5), 557–570 (2002)

An Evidential and Context-Aware Recommendation Strategy to Enhance Interactions with Smart Spaces

Josué Iglesias, Ana M. Bernardos, and José R. Casar

Universidad Politécnica de Madrid, Telecommunications School, Madrid, Spain
{josue,abernardos,jramon}@grpss.ssr.upm.es

Abstract. This work describes a novel strategy implementing a context-aware recommendation system. It has been conceived to offer an intelligent selection of micro-services used to orchestrate networks of smart objects taking into account users' needs and preferences. The recommendation offering dynamically evolves depending on users' micro-service management patterns and users' context. The complete system has been designed within Dempster-Shafer evidential theory framework, ensuring uncertainty support both at context acquisition and at recommendation configuration level.

Keywords: Dempster-Shafer evidential theory, context-aware services, recommendation systems, smart spaces, smart objects.

1 Introduction

The concept of *smart space* is becoming popular to describe intelligent environments able to satisfy their inhabitant needs. In brief, smart spaces can be considered as a set of coordinated smart objects coexisting in the same environment. Our previous works (e.g., [1]) address smart objects coordination in an environment where (i) smart objects are able to publish their capabilities and (ii) users may configure cooperative smart object *behaviours*. These *behaviours* are constructed and evaluated from the user's personal mobile device in the form of Event-Condition-Action (ECA) rules (e.g., 'IF *I'm approaching home* AND *no one is there* THEN *turn the heater on*'). This smart object orchestration approach empowers the user to configure his/her personal set of ECA rules, enabling the emergence of a 'shared behaviours market'. This proposal faces several challenges, some of them related to the delivery of a satisfactory user experience. In particular, in a near future, smart spaces may be densely populated by a great number of smart objects, each of them offering several capabilities and with different ways of combining them. This potential environment may overwhelm the user when trying to personalize a smart space and encourages the appearance of new techniques to filter the available information and adapt it to the user's particular needs.

Thus, this paper proposes a context-based information filtering mechanism to enhance interactions with smart spaces. It specifically proposes a novel strategy

for recommending already developed *behaviours* (i.e., ECA rules) used to orchestrate networks of smart objects. Dempster-Shafer evidential theory (DSET from now on) capabilities for handling uncertainty and ignorance are exploited in order (i) to model user context acquisition process, (ii) to map user context and *behaviours* to recommend and (iii) to quantify *behaviours* usage patterns. A contextual update strategy is also proposed in order to dynamically adapt the recommendation offering according how the users consume those *behaviours*.

Section 2 reviews the relation between DSET and context-aware recommendation systems. An overview of DSET and how it can be exploited for recommendation purposes is the focus of Section 3. The recommendation mechanism is deeply described in Section 4. Section 5 focuses on the contextual update of the recommendation. Finally, Section 6 analyses some preliminary validation tests and Section 7 offers some conclusions and anticipates future works.

2 Related Research

Regarding the techniques for supporting recommendation, and beyond the classical differences between content-based and collaborative recommenders, the relatively new field of context-aware recommender systems is deeply covered in [2], where several techniques are mentioned for implementing model-based recommendations, i.e., predictive models for calculating the probability with which the user chooses a certain type of item in a given context (e.g., support vector machines or Bayesian classifiers). As a generalization of the Bayesian probability theory, DSET extends uncertainty support, e.g., by explicitly representing ignorance in the absence of information, by offering a simple mechanism for evidence propagation or by a limited reliance on training data [3][4]. However, it is difficult to find in the literature references to systems implementing DSET mechanisms for supporting recommendations. It is necessary to search within the decision making area in order to find researches implementing DSET-based intelligent selections mechanisms. Most of them are based on payoff matrices, built by experts, linking several states of nature to different alternatives and where the knowledge of the state of nature is captured in terms of a belief mass function (a DSET concept explained in Section 3.1) [5]. Based on this idea, our work also proposes to model the quantification of the relation between that state of nature (*context* in our case) and the alternatives (e.g., *items* to recommend) adopting the concept of evidential mappings: an extension of the DSET where belief mass functions are used to represent uncertain relationships [6]. Evidential mappings have been used for location and activity estimation (e.g., [3][4]) but, as far as we are concerned, no research has been conducted in order to exploit this technique for recommendation purposes.

3 Motivation: Enhancing Smart Space Personalization

Fig. 1 outlines a generic smart space to be personalized by means of the definition of ECA rules working on the capabilities offered by the sensors integrated into

different smart objects. The user is able to deal with different kinds of entities in his/her daily life: physical objects (smart or not), services (real or virtual) and other people. Beyond dynamic context information acquired from sensors, semi-static information about the user is also available in the form of a personal profile.

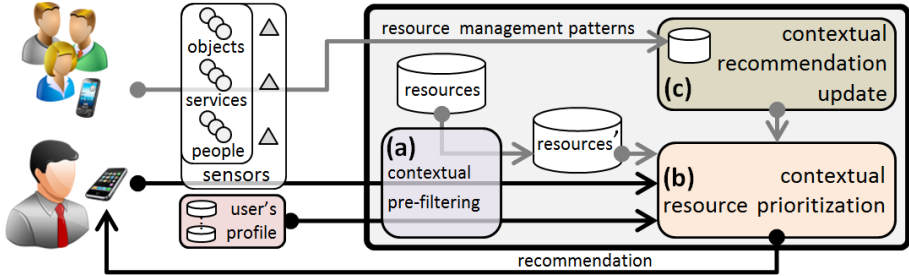


Fig. 1. Evidential recommendation service overview

Both, the information acquired from sensors and that stored in the personal profile are inputs of a recommendation system aiming at making a context-based selection of already developed ECA rules used to personalize the smart space (let denote these set of rules as 'resources'). The recommendation process involves two different phases: (i) a contextual pre-filtering mechanism (Fig. 1.a; not covered in this paper) implementing an intelligent selection of the resources to take part in (ii) a multidimensional recommender in charge of making a contextual prioritization of resources (Fig. 1.b). The relation between the user context and the resources to recommend is dynamically built and constantly updated taking into account the context of the users when manipulating (create, execute, share, delete, (de)activate, download or modify) the resources (Fig. 1.c).

Basically, the proposed system has to deal with uncertain information when handling the information acquired from sensors (inherently uncertain entities with some reliability associated [7]) and when defining the relation between context and resources to recommend (which an expert may not be unequivocally certain about). Thus, the recommendation system has been built following a DSET approach, whose capabilities for handling uncertainty and ignorance are next detailed.

3.1 Dempster-Shafer Evidential Theory

DSET was originally developed from Dempster's research and later completed by Shafer [8]. It offers a mathematical method for handling subjective beliefs (evidences) over a set of hypotheses $\Omega = \{h_1, h_2, \dots, h_N\}$, called *frame of discernment*, that has to be exhaustive and with mutually exclusive elements.

Uncertainty assignment is performed in DSET by means of a *belief mass function* $m(\cdot)$. This distribution can assign evidence to any combination of

elements in Ω , i.e., $m : 2^\Omega \rightarrow [0, 1]$. It should also satisfy that $m(\emptyset) = 0$ and $\sum_{\forall A_i | A_i \in 2^\Omega} m(A_i) = 1$.

$m(A)$, with $A \in 2^\Omega$, represents the proportion of all relevant and available evidence that supports the claim that the hypothesis A is true, offering no information about the evidence assigned to any subset of A . Evidence assigned to singletons constitutes more precise knowledge than evidence assigned to other subsets of Ω .

A belief mass function $m(\cdot)$ on the frame of discernment Ω generates two other set functions also defined on 2^Ω : *belief* $Bel(\cdot)$ and *plausibility* $Pls(\cdot)$. $Bel(\cdot)$, defined as $Bel : 2^\Omega \rightarrow [0, 1]$, is a measure of the (total) evidence certainly assigned to a hypothesis (e.g., A). It represents our confidence that A or any subset of A is true: $Bel(A) = \sum_{\forall B_i | B_i \subseteq A} m(B_i)$. $Pls(\cdot)$ is also defined as $Pls : 2^\Omega \rightarrow [0, 1]$. It is a measure of the evidence that could be possibly assigned to A , that is, evidence assigned to any hypothesis consistent with A (i.e., any hypotheses not contradicting A): $Pls(A) = \sum_{\forall B_i | A \cap B_i \neq \emptyset} m(B_i)$. Some authors (not everyone) interpret $Bel(\cdot)$ and $Pls(\cdot)$ functions as a kind of lower and upper bounds of a probability function (in fact, the interval between these two functions is known as *belief interval* [$Bel(\cdot), Pls(\cdot)$]).

DSET also provides a method to combine the measures of evidence from independent sources: the *Dempster's rule of combination* [9]:

$$(m_1 \otimes m_2)(A) = \frac{\sum_{\forall B, C | B \cap C = A} m_1(B) \cdot m_2(C)}{1 - \sum_{\forall B, C | B \cap C = \emptyset} m_1(B) \cdot m_2(C)} \tag{1}$$

3.2 Evidential Mappings

Elements of different frames of discernment can be related through an *evidential mapping*, i.e., a causal link among elements of two frames in the form of mass functions. An evidential mapping Γ^* from frame Ω_E (representing known evidences) to frame Ω_H (representing hypothesis to calculate) is called a Complete Evidential Mapping (CEM) if it assigns to each subset of Ω_E a set of 'subset-mass pairs' from Ω_H (i.e., $\Gamma^*(E_i) = \{(H_1, g(E_i \rightarrow H_1)), \dots, (H_M, g(E_i \rightarrow H_M))\}$). A deep analysis regarding evidential mappings can be found in [6]. Then, a piece of evidence on Ω_E can be propagated to Ω_H through an evidential mapping as follows:

$$m_H(H_j) = \sum_{j=1}^M m_E(E_i) \cdot g(E_i \rightarrow H_j) \tag{2}$$

Next Section explores how evidential mappings are exploited in order to support sensor-based context acquisition and contextual recommendation.

4 Recommendation Service Description

4.1 Sensors Evidential Modelling

In general, sensors are to be modelled by means of Γ_i^S CEM. It relates the evidences Ω_{S^i} a sensor offers over a context variable and the real status of that variable Ω_{V^i} (index i identifies each sensor). In this work sensors are considered to be evidential, i.e., they estimate reality in the form of a belief mass function $m_{S^i}(\cdot)$, that can be used in (2), together with Γ_i^S , in order to calculate $m_{V^i}(\cdot)$.

Example 1. Table 1.a exemplifies a CEM modelling the estimates obtained from a location system with 3 possible symbolic locations $\{a, b, c\}$. For instance, Γ_{loc}^S states that 'if the location sensor estimates that the user is located at "c", then the user is actually located at "b" or "c" with an evidence of 0.1 and ...'. It is worth noting that this sensor modelling includes ignorance modelling at evidence level in the form of the belief mass assigned to combinations of the singletons within Ω_{S^i} .

Table 1. Example of CEMs modelling (a) a location sensor and (b) a context-resource recommendation

(a)		context hypotheses $\Omega_{V^{loc}}$								
		$m_{S^{loc}} \Gamma_{loc}^S$	\emptyset	a	b	c	ab	ac	bc	abc
sensor evidences $\Omega_{S^{loc}}$	0.0	\emptyset	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.8	a	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.5
	0.0	b	0.0	0.1	0.6	0.0	0.1	0.0	0.0	0.2
	0.0	c	0.0	0.0	0.0	0.7	0.0	0.1	0.1	0.1
	0.0	ab	0.0	0.1	0.1	0.0	0.8	0.0	0.0	0.0
	0.0	ac	0.0	0.2	0.0	0.1	0.0	0.7	0.0	0.0
	0.0	bc	0.0	0.0	0.0	0.0	0.0	0.9	0.1	0.0
	0.2	abc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
			$m_{V^{loc}}(6)$	0.0	0.4	0.0	0.0	0.0	0.0	0.0

(b)		resource prioritization hypotheses Ω_R									
		$m_{V^{loc}} \Gamma_{loc}^R$	\emptyset	r_1	r_2	r_3	r_1r_2	r_1r_3	r_2r_3	$r_1r_2r_3$	
context evidences $\Omega_{V^{loc}}$	0.0	\emptyset	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	0.4	a	0.0	0.0	0.9	0.0	0.0	0.0	0.0	0.1	
	0.0	b	0.0	0.0	0.2	0.0	0.4	0.0	0.2	0.2	
	0.0	c	0.0	0.1	0.0	0.3	0.6	0.0	0.0	0.0	
	0.0	ab	0.0	0.5	0.0	0.0	0.0	0.5	0.0	0.0	
	0.0	ac	0.0	0.0	0.3	0.0	0.3	0.3	0.0	0.1	
	0.0	bc	0.0	0.0	0.0	0.1	0.0	0.0	0.9	0.0	
	0.6	abc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	
			$m_{R^{loc}}(6)$	0.0	0.0	0.36	0.0	0.0	0.0	0.0	0.64

4.2 Evidential Decision Making

Context-Resource Evidential Mapping. An evidential decision making process, aiming at offering a context-prioritized list of resources, is also built from another set of CEMs (Γ_i^R). In this case, the evidential mapping links each context variable Ω_{V^i} modelled according a belief mass function obtained from the above mentioned sensor modelling process (m_{V^i}) with a common frame of discernment $\Omega_R = \{r_1, r_2, \dots, r_N\}$ representing resources to recommend.

Once again, (2) can be used in order to calculate $m_{R^i}(\cdot)$, i.e., the partial belief mass function representing evidences regarding how to prioritize the resources taking only into account the context provided by $m_{V^i}(\cdot)$.

Example 2. Table 1.b shows Γ_{loc}^R , the CEM representing the relation between the possible locations of a user ($\Omega_{V^{loc}}$) and the resources to be prioritized (only 3

resources are considered in this example: $\{r_1, r_2, r_3\}$). For instance, it states that *'if the user is located at "c", then the resources "r1" or "r2" should be recommended with an evidence of 0.6 and ...'*. This example also covers ignorance modelling at mapping level, this time in the form of evidence assigned to $g(E_i \rightarrow \Omega_R) \forall i$ mappings.

Evidential Fusion. At this point, partial information on how to distribute the resource recommendation taking into account different types of context (i.e., the complete set of partial $m_{R^*}(\cdot)$ belief assignment functions) is aggregated using Dempster's rule of combination (1) obtaining $m_{R^*}(\cdot)$.

Evidential Prioritization Strategy. $m_{R^*}(\cdot)$ can be considered as a score rating the suitability of each resource (or set of resources) taking into account the complete set of available context. Remembering the definition of belief mass function from Section 3.1, it has to be noted that the complete suitability assigned to a resource, e.g., r_i , is not included just in $m_{R^*}(\{r_i\})$, but also (partially) in the belief mass assigned to other subsets of Ω_R , e.g., in $m_{R^*}(\{r_i, r_j\})$, $m_{R^*}(\{r_i, r_j, r_k\})$, etc. $Bel(\cdot)$ and $Pls(\cdot)$ functions provide complementary approaches for calculating the complete resource recommendation suitability in the form of a belief interval $[Bel_{R^*}(\cdot), Pls_{R^*}(\cdot)]$.

Although other techniques do exist, resource recommendation has been developed in this work applying a Minimax Regret Approach (MRA) [10] to the set of belief intervals describing each resource. MRA assures optimality in a worst case scenario, being able to detect the resource that minimizes the maximum difference of expected evidence among the complete set of resources (3)(4).

$$q(r_i) = \max_{\forall j \neq i} [Pls_{R^*}(r_j)] - Bel_{R^*}(r_i) \tag{3}$$

$$Q(\Omega_R) = \arg \min_{\forall i} [q(r_i)] \tag{4}$$

The iterative algorithm in Fig. 2 exploits (3) and (4) in order to obtain an ordered ranking of resources. \overline{Q}_{R^*} vector, initially empty, represents the ordered list of resources to be calculated. The algorithm iteratively applies MRA (3)(4) to Θ . Although Θ is initially composed by the complete set of resources ($\Theta = \Omega_R$), the most suitable resource calculated $Q(\Theta)$ is removed from Θ at each iteration in order to apply MRA only to the rest of resources and then iteratively calculate the recommendation order.

5 Evidential Mapping Contextual Update

Besides acquiring user context for recommendation purposes, sensor data can be used to quantify user's patterns in resource management. This resource management information can be exploited in order to dynamically update CEMs modelling context-resource mappings Γ_i^R , being then able to offer recommendations correlated with the real manipulation of resources.

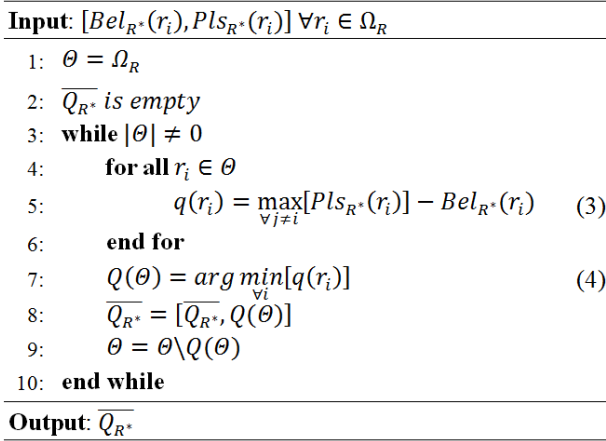


Fig. 2. Belief interval based recommendation algorithm

Γ_i^R update is dynamically computed taking into account single user's resources management. Then, the particular behaviour of individual users regarding resource management (and his/her particular context) is used to update the global recommendation used for every user (Γ_i^R). Information regarding resource management operations is also modelled as evidential information obtained from in-device sensors installed in the user mobile device.

Matrix M_{jk}^i stores evidential information on how user u_i manages resource r_j at a specific moment. Individual resource management is modelled by assigning evidences over the frame of discernment covering the complete set of management operations $\Omega_L = \{l_1, \dots, l_L\}$ (k index in M_{jk}^i is used to reference each element in 2^{Ω_L}), so $\sum_{\forall k} M_{jk}^i = 1$. Each time M_{jk}^i is modified (i.e., each time a particular user manipulates in any sense a resource) user context would be also stored in S_{jk}^i matrix. S_{jk}^i assigns evidences over Ω_{V^j} , i.e., evidences supporting the fact that context variable V^j is in state s_k for user u_i ($\sum_{\forall k} S_{jk}^i = 1$).

Then, in order to contextualize resources management, the joint (i.e., multidimensional) belief mass function U_{jkmn}^i is constructed using (5), assigning evidences over the product frame $\Omega_{U^m} = \Omega_L \times \Omega_{V^m}$. U_{jkmn}^i represents a way of quantifying how a particular management operation $k \in 2^{\Omega_L}$ over a resource r_j is distributed among the different states s_n of different context variables V^m for a given user u_i (with $\sum_{\forall k,n} U_{jkmn}^i = 1$). Finally, resource management information, stored in different U_{jkmn}^i matrices (one per user), is aggregated using (6), also verifying that $\sum_{\forall k,n} U_{jkmn}^T = 1$.

$$U_{jkmn}^i = M_{jk}^i \cdot S_{mn}^i \tag{5}$$

$$U_{jkmn}^T = \frac{1}{N_U} \sum_{\forall i} U_{jkmn}^i \tag{6}$$

Γ_i^R dynamic update is performed applying the corresponding update factor α_{ijk} to each of its elements each time \mathbb{U}_{jkmn}^T is modified, i.e., $g_{\Gamma_i^R}(j \rightarrow k) = \alpha_{ijk} + g'_{\Gamma_i^R}(j \rightarrow k)$, where $g'_{\Gamma_i^R}(j \rightarrow k)$ represents the value assigned to each element in Γ_i^R before recommendation update. α_{ijk} is obtained from \mathbb{U}_{jkmn}^T by means of (7).

$$\alpha_{ijk} = \sum_{m=1}^{|2^{\Omega_L}|-1} x_m \cdot \mathbb{U}_{kmi}^T \tag{7}$$

x_m in (7) are integer values (positives or negatives) associated to each element in 2^{Ω_L} ; they are used to quantify to which extend each kind of resource manipulation should make $g_{\Gamma_i^R}(j \rightarrow k)$ evolve from its previous value $g'_{\Gamma_i^R}(j \rightarrow k)$. Thus, (7) aggregates the effect of different management operations in the recommendation into a single value (α_{ijk}). It is easy to see that $\alpha_{ijk} > 0$ leads to increasing $g_{\Gamma_i^R}(j \rightarrow k)$, $\alpha_{ijk} < 0$ results in decreasing it and no update in the recommendation is obtained for $\alpha_{ijk} = 0$.

6 Recommendation Update: Tests and Evaluations

In order to check the contextual update of the recommendation, a simulation scenario has been built. It is composed of 10 users able to perform 2 different management operations ($l_2 = \{download\}$ and $l_3 = \{delete\}$) over a set of 3 resources (i.e., 3 different ECA rules configuring each of them some kind of *behaviour* for the smart space). The recommendation is updated taking into account 2 context variables representing 7 symbolic locations and 4 temporal parts of the day (*morning, afternoon, evening* and *night*) respectively.

Starting from a random recommendation ($\Gamma_i^R; \forall i$) and contextual usage matrix ($\mathbb{U}_{jkmn}^i; \forall i$), users perform different management operations over several resources each Δt ; all these operations are performed in the same context in order to test how the recommendation evolves (see Fig. 3's configuration table for simulation details).

Fig. 3.b depicts a scenario where users tend to perform operation $l_2 = \{download\}$ over resource r_3 . It verifies that, for the particular context of 'being located in roomA', this resource increases its recommendation (i.e., $g_{\Gamma_{loc}^R}(A \rightarrow r_3)$) as $x_2 > 0$. The new mass assigned to r_3 recommendation leads to proportionally decrease the mass assigned to non-singletons values in Ω_R . A zoom is also presented in order to highlight Δt and context-dependent variable α_{ijk} . Equivalently, Fig. 3.c represents an intensive use of operation $l_3 = \{delete\}$ over r_1 . As this operation has associated a negative impact on the recommendation ($x_3 < 0$), then the recommendation score associated to r_1 is decreased. In this case this decrement is compensated by increasing total ignorance (Ω_R). Fig. 3.d represents other operation-resource pair as defined in Fig. 3's configuration table. No operations are held in Fig. 3.a and Fig. 3.e.

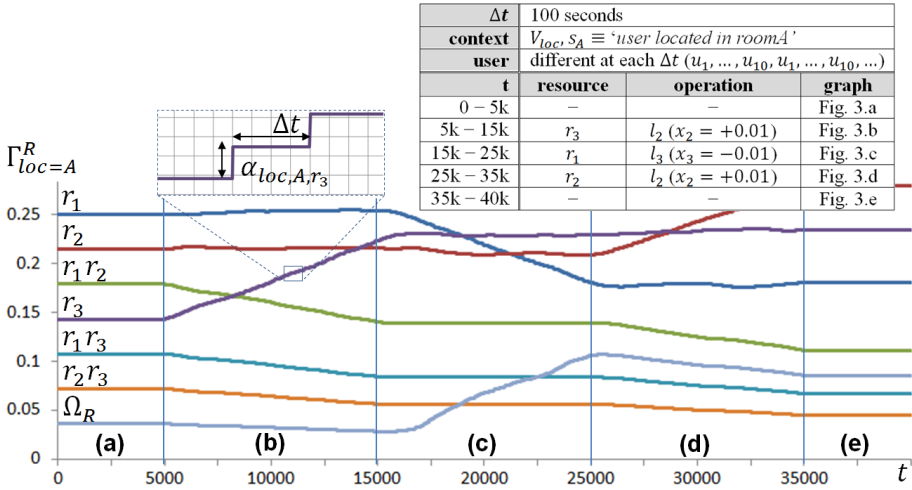


Fig. 3. Recommendation update simulation configuration details and graph

7 Conclusions and Future Works

This paper describes a novel strategy implementing context-aware recommendations of micro-services for smart spaces personalization. Both, the phase designed to calculate micro-services priority and the one in charge of updating the recommendation according user's micro-service management patterns are supported by DSET in order to ensure uncertainty support at different levels. Simulation tests have been executed in order to functionally validate the strategy.

Some future works are already planned for enhancing the recommendation process. For example, in this work just instantaneous events are considered as possible management operations to be applied to the resources and we are already working on also being able to deal with other types of operations involving temporal durations. Besides, recommendation update for a particular management operation is quantitatively equal for any kind of context, but future extensions may consider context-dependant update factors (i.e., making x_m in (7) contextual). In a more abstract perspective it can be argue that neither context nor resources to recommend are in this work related; hierarchically modelling these entities (using semantic technologies, for instance) may lead to improvements in the recommendation [2] (e.g., instead of recommending single resources, it could be possible to recommend types of resources). Furthermore, the system presented may be considered user-context-driven in the sense that only the context of the user is the one able to modify the recommendation, but resources also have their own context (e.g., expiration date) and then new functions for modifying the recommendation may appear based on this fact. Another issue to enhance in the update recommendation process is related to the fact that it always leads in decrementing belief mass associated to combination of resources (except for total ignorance Ω_R ; see r_1r_2 , r_1r_3 and r_2r_3 in Fig. 3) and only a

system administrator may change this tendency. Based on the idea of Shafer's partition technique [6], we are already working in a new definition of M_{jk}^i in order to let the system change this kind of uncertainty automatically.

Finally, we are also planning to deploy this recommendation service in a real scenario. In this sense, it would be interesting to apply it to solve other recommendation problems within the smart space domain (e.g., for supporting intelligent selection of interfaces).

Acknowledgments. This work has been supported by the Spanish Ministry of Economy and Competitiveness through the CDTI CENIT THOFU Programme and by the Government of Madrid under grant S2009/TIC-1485 (CONTEXTS).

References

1. Bernardos, A.M., Casar, J.R., Cano, J., Bergesio, L.: Enhancing interaction with smart objects through mobile devices. In: Proceedings of the 9th ACM International Symposium on Mobility Management and Wireless Access, MobiWac 2011, pp. 199–202. ACM, New York (2011)
2. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: ACM Conference on Recommender Systems, pp. 335–336 (2008)
3. Hong, X., Nugent, C., Mulvenna, M., McClean, S., Scotney, B., Devlin, S.: Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive and Mobile Computing* 5(3), 236–252 (2009)
4. McKeever, S., Ye, J., Coyle, L., Dobson, S.: Using dempster-shafer theory of evidence for situation inference. In: Barnaghi, P., Moessner, K., Presser, M., Meissner, S. (eds.) EuroSSC 2009. LNCS, vol. 5741, pp. 149–162. Springer, Heidelberg (2009)
5. Casanovas, M., Merigó, J.M.: Fuzzy aggregation operators in decision making with dempster shafer belief structure. *Expert Systems with Applications* 39, 7138–7149 (2012)
6. Liu, W., Hughes, J.G., McTear, M.F.: Advances in the dempster-shafer theory of evidence, pp. 441–471. John Wiley & Sons, Inc., New York (1994)
7. Tolstikov, A., Hong, X., Biswas, J., Nugent, C., Chen, L., Parente, G.: Comparison of fusion methods based on dst and dbn in human activity recognition. *Journal of Control Theory and Applications* 9, 18–27 (2011)
8. Shafer, G.: *A Mathematical Theory of Evidence*. Princeton University Press, Princeton (1976)
9. Sentz, K., Ferson, S.: Combination of evidence in dempster-shafer theory. Technical report, Sandia National Laboratories, SAND 2002-0835 (2002)
10. Savage, L.J.: The theory of statistical decision. *Journal of the American Statistical Association* 46(253), 55–67 (1951)

Information Fusion for Context Awareness in Intelligent Environments

Fábio Silva, Cesar Analide, and Paulo Novais

University of Minho

{fabiosilva, analide, pjon}@di.uminho.pt

Abstract. The development of intelligent environments requires handling of data perceived from users, received from environments and gathered from objects. Such data is often used to implement machine learning tasks in order to predict actions or to anticipate needs and wills, as well as to provide additional context in applications. Thus, it is often needed to perform operations upon collected data, such as pre-processing, information fusion of sensor data, and manage models from machine learning. These machine learning models may have impact on the performance of platforms and systems used to obtain intelligent environments. In this paper, it is addressed the issue of the development of middleware for intelligent systems, using techniques from information fusion and machine learning that provide context awareness and reduce the impact of information acquisition on both storage and energy efficiency. This discussion is presented in the context of PHESS, a project to ensure energetic sustainability, based on intelligent agents and multi-agent systems, where these techniques are applied.

Keywords: Information Fusion, Machine Learning, Intelligent Environments, Context Awareness.

1 Introduction

Ubiquitous spaces are a common research field nowadays, mostly due to the increase in sensors installed on environments and the technological opportunity it presents. This, coupled with the recent surge of ubiquitous devices and applications, has led to the opportunity to create budget-friendly intelligent environments with different objectives, ranging from energy efficiency, sustainability and user comfort [1], [2], acquiring user context, assisted living and automate tasks [3], [4]. Such environments are able to monitor users, objects and the environment itself, generating rich sets of data, upon which may be used to make decisions and perform optimizations.

In a wide range of practical applications, information is obtained not only from processing data acquired through sensors in a ubiquitous environment, but also from information and knowledge shared across environments, such as mathematical models, profiling and machine learning models. Contexts can be created by fusing data from sensors and other sources of information. Such concept is designated as information fusion and is used for tasks that involve gathering information from different

sources, using it to improve its quality, accuracy or derive new information [5]. An example of this approach can be found in the Sensor 9K testbed [6] where data about humidity, temperature, air velocity, among others, is used to derive human thermal sensation. Such sets of data, information and knowledge can be used as context to help identify profile and optimize solutions and may be accessed directly through sensor data, middleware or context servers [7].

Considering a system designed to save energy, from a sustainable perspective [8], it entails a delicate equilibrium due to the fact that any effort made in order to gather knowledge incurs in energy expenditure and, thus, this expenditure needs to be significantly lower than the saving obtained with the information gathered. In terms of research methodology, this is usually called the observer dilemma, where the observation, by itself, introduces changes to the actual state of the system. In energetic terms, observing energy consumption and computing energy saving measures increases consumption, thus changing the problem in the process, creating an overhead that needs to be mitigated by the end solution.

Hybrid structures and planning are often means to reduce the impact of certain solution upon the global objective. One strategy to tackle this problem is to use shared data between different systems to their benefit. However, this solution needs to generalize assumptions about environments and thus reduce optimization opportunities in specialized environments. Some approaches use only user awareness by using monitoring sensors that transmit current data about consumption and aggregation systems. Other hybrid approaches use both generalizations with some contextual specializations in order to introduce some context to the solutions represented.

With information fusion, context awareness and machine learning models, it is proposed a set of strategies aiming to reduce energy and storage expenditure, reduce side effects from this workflow while maintaining accuracy and context-aware capabilities. Such strategies were used in the PHESS project, currently being developed with the aim to bring energy efficiency, comfort and sustainability to intelligent environments.

1.1 Information Fusion

Information fusion comprises the use of heterogeneous and homogenous data and information sources. There is some confusion with the terminology as some authors use the terms sensor fusion, data fusion and information fusion with the same meaning [8]. Nevertheless, it is commonly accepted that sensor fusion is a subdomain of information and data fusion as it only considers the use of data from sensors. With these theories it is possible to maintain and update information, enrich data creating new content, improving quality and providing more accurate contexts. Information fusion might also be used in order to enrich with additional contexts machine learning models describing environments, behaviors and actions inside intelligent systems. It offers basic steps for data pre-processing in machine learning activities, but they are also used to build data models and extract information [5]. Data by itself is limited in the type of knowledge and information that can be extracted from such environments. Analyzing data from different sources poses the opportunity to increase the quality of the measurements, although it may also increase some uncertainty as well [9].

Multi-sensor management and sensor fusion are terms applied when the source of data are sensors and it is defined by Xiong and Svensson [10] as a process that manages and coordinates the use of a number of sensors in a dynamic uncertain environment with the aim to improve data fusion. Sensor fusion tasks have to take in consideration a number of factors such as data imperfection and outliers, conflicting data, data modality, data correlation, data alignment, data association, processing framework, operational timing, static versus dynamic phenomena and data dimensionality [11]. In order to tackle data imperfection a number of filters and inferences were developed such as Bayesian inference, probabilistic grids, Kalman filters and Monte Carlo methods.

1.2 Machine Learning

Machine learning techniques allow the modeling and learning of preferences and habits in different contexts. These techniques also allow the learning of past and current trends and predict future results. Among the contexts where the use of machine learning provides an opportunity to enhance systems, there is the concept of sustainability. With information from one or several environments, machine learning theory can derive models of behavior and interaction based on specialized contexts.

Machine learning and data mining techniques can also be used to obtain information about user's habits in intelligent environments. In this aspect, there are re-search examples demonstrating several algorithms that perform this task from data gathered by sensors in the environment. These algorithms use theory from Sequence Discovery, Fuzzy Logic, Genetic Programming, Multi-Layer Perceptron and combinations of these techniques [12]. Other uses for machine learning is the discovery of rule sets to monitor and man-age the consumption of resources such as energy inside intelligent environments [13].

1.3 Context Awareness

Context-aware systems are a component of ubiquitous computing or pervasive computing environments. These systems consider information about location, environments, resources, users and relationships between each concept. It hopes to make informed and personalized decisions, based on contextual factors that might promote distinct decisions in similar situations with different contexts [14]. Universal models and information provide explanation about phenomena that may be accurate and useful. However, when there is a need to specialize to certain contexts, it may be needed to detail models making them more accurate according to a known context. On the other hand, their specialization often reduces their generalization which becomes a trade-off between increased accuracy and generalization. Nevertheless, methodologies employed may be repeatable among different environments even though they do not generate the same model for the same attribute. Context aware elements can be acquired by the use of information present in intelligent environments through direct sensor access, middleware infrastructure or, even, context servers, as detailed by Baldauf, Dostar and Rosenberg [7]. Sensors used in context aware systems may be

classified in three groups: physical sensors, virtual sensors and logical sensors. Physical sensors refer to context gathered by physical devices sensing the environment. Virtual sensors are defined by the use of application and services as sources of contextual data. Logical sensors combine physical and virtual sensors to determine logical values for the attribute being sensed.

Elements of context are often gathered using all three types of sensor classification according to contextual nature inside intelligent environments. Strategies for the management of context models can be defined as Key-value models, Markup schemes, Graphical Models such as UML, Object Oriented models, Logic Based Models and Ontology based Models. Context in PHESS project is defined by sensor data models from the environment and status indicators in terms of sustainability indexes. Each of these factors provides important information saved in terms of context, towards the application of contextualized options in the PHESS project. Model development through machine learning is the methodology used to store information about sensors in the environment. So, with the help of sensor data, models, and user presence and sustainable indexes it is possible to assess the impact of users inside environments in a contextualized analysis.

1.4 Intelligent Environments

Intelligent Environments with applications towards user assisted living are already under study and object of discussion by the research community. Focus has been applied to the study of behaviors, routines, stress assessment, energy efficiency and task prediction. Ubiquitous environments present a significant opportunity for learning tasks and contextualized optimizations. The data and information shared between intelligent objects, environments and users entails a delicate balance that must be taken into consideration when assessing human comfort condition and planning interventions on the environment. Some implementations of intelligent environments are used to perform experiments on ambient intelligence theory. iDorm is an examples of such scenario, where sensors can gather data about temperature, occupancy, humidity, and light levels. The actuators can open and close doors, and adjust heaters and blinds. Other example is HomeLab [15] composed of a house filled with hidden cameras, microphones and a remote power control system able to operate switches and control lightning. This lab is used by researchers to assess social responses to different color schemes in lightning and monitor its users. Yet, another intelligent environment can be found in MavPad project, which uses a smart apartment created within the project [3]. This project consists of a living/dining room, a kitchen, a bathroom, and a bedroom, all fitted with different types of sensors to gather information from objects, users and contexts. Saves is a project that encompasses an intelligent environment designed to use building and user occupancy profiles to maintain and regulate temperature inside a building [1]. Sensor 9k acts as an intelligent environments middleware for creating and promoting intelligent environment applications [6].

The testbed I3A is composed by a sensor network displayed through a building sensing information about temperature, humidity, carbon, carbon dioxide, dust and electrical appliances [16]. This testbed is used to prototype solution for intelligent

systems as each sensor node can be independently programmed in the wireless sensor network covering the building.

The approach taken for intelligent environments embedded in this work use concepts shared from the environments and platforms already mentioned such as sensor network, profiling and sensorization of users and environments. Nevertheless, focus has been made in creating and maintaining machine learned models reduce dependence on constant sensorization and ease the data storage effort while providing context-aware computing.

2 PHESS – People Help Energy Savings and Sustainability

The PHESS project (People Help Energy Savings and Sustainability) is an integrated system to monitor and reason about environments and users with the objective of helping users save energy and ensure sustainability as well as their own comfort [17]. This system makes use of sensor networks, spread both on environments and users, acquiring data about user actions, environment variables and environment status to deliver a contextualized analysis. The PHESS project uses a layered architecture in which are included layer for sensors, models, reason. Each of these layers is responsible for a segment on the system's operation.

2.1 Sensor Layer

Currently, the PHESS project is able to integrate results from different sensors, using a multi-agent architecture where agents publish sensor interfaces for the consumption of other sensors. Sensor fusion is obtained creating virtual sensors that, instead of relying in physical hardware to provide sensor data, rely on the consumption of sensors already present in the platform which are processes according to the sensor fusion strategy in place. Sensor fusion is then obtained from specific virtual agents launched in the platform. These algorithms will be used in order to mitigate some characteristics of the devices: sensor inaccuracy, false readings or conflicting data. These virtual agents are the responsible for data fusion, creating new variables, such as thermal sensation and user occupancy.

2.2 Model Layer

The main goal of the layer dedicated to models is to reduce both the energetic impact of the sensing platform and the traffic flow, and, at the same time, optimize the general system response. Models are able to characterize behavior, anticipate and predict values for the attribute being studied and do so efficiently if properly built. Leveraging these properties it is possible to use models locally instead of demanding complex operations on databases such as aggregations and large quantities of storage space to record historic activity.

This layer is used to create models about environment, environment variables and user habits and preferences. Models are defined by intelligent agents present on the PHESS project. According to the type of model, they may require information and data from other agents.

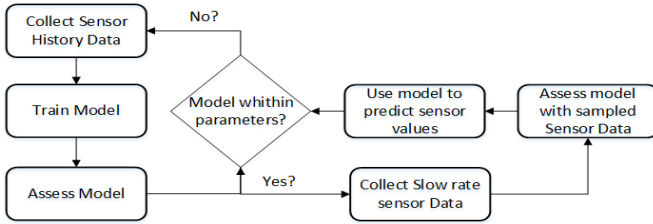


Fig. 1. Model Management

One type of models used encompasses mathematical models which use knowledge about attributes defined by mathematical and physic rules. Another type uses and combines data and information from other models on mature and accepted models that may also be described by mathematical rules. Lastly, from the data continuously gathered from the data layer, models that mimic the behavior of those attributes in order to provide description of their behavior and provide means to anticipate or predict the future state of the environment. These models require a constant validation in order to assess the validity of the models created in each environment. Also, due to possible high number of sensors and costs related to storage of records and historic data, these models may be used as historic descriptive models and as a sensor alternative in order to save in traffic messages as detailed in figure 1.

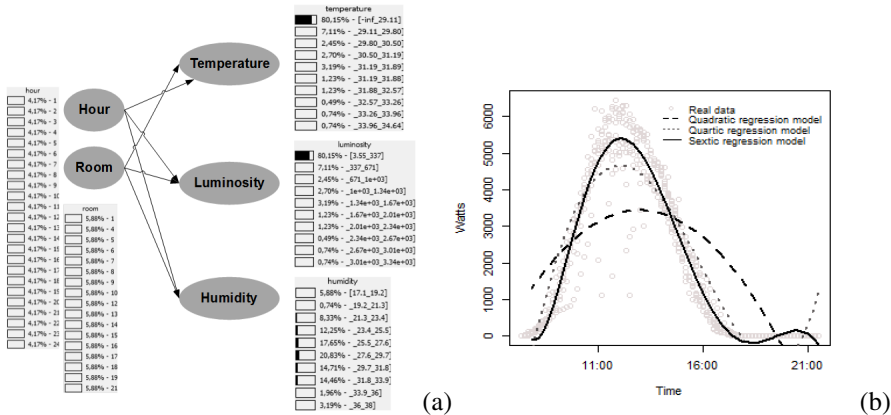


Fig. 2. Bayesian Network model describing an environment temperature, luminosity and humidity (a), Electrical Photovoltaic regression model (b)

Machine learning acts as a methodology to estimate sensor readings and, while doing so reduce the need to high sample rates in the sensor layer. With the combination of initial learning models and constant validation of its accuracy and significance in the system, refresh rates for sensor values can be dynamically managed. The usage of these schemes may also be relevant for sensor fusion tasks, since modeled sensors may be directly assessed in the server side of the platform, leaving the client side less demanding in terms of computational effort. As examples of agents in the model layer

it can be considered electrical consumption, temperature, luminosity and solar exposure agents. Figure 2 (a) presents the representation of a Bayesian network model that stores a grid of conditional probabilities for the value of temperature, luminosity and humidity in an environment, according to the time of day and room. Figure 2 (b) details a regressive model to estimate photovoltaic electric energy production, according to weather (mostly cloudy in the case depicted) and time of day.

2.3 Reasoning Layer

The reasoning layer uses automated reasoning workflows, as well as on-demand simulation tasks taking the models created for the environment as reference and input variables. Examples of workflow methodology include automated case-based reasoning to find possible optimizations in terms of appliances, and behaviors through profile comparison. A first approach to reason about alternative solutions in the environment considers the use of case-based reasoning. In this approach, current models and environment specifications are compared to other known implementation and solutions in order to quickly assess optimized solutions for the environment. The static components in the environment, such as appliances or lightning bulbs, can be quickly assessed in order of efficiency in terms of energy consumption efficiency and whether changing them is beneficial to the overall process of energy optimization [18].

The final step of action of this agent is to use the newly calculated situations and use actuator agents to enforce the new plan or when such is not possible, send a report to the user so he can become aware of efficient changes in the environment without affecting its interaction with the environment.

3 Context Awareness in PHESS

The creation of context for intelligent agents on this platform is done as described in a survey on context-aware systems by Baldauf, Dostar and Rosenberg [7]. In detail, it is considered a context server, middleware and direct sensor access as a source of context. The context server is provided by an application server running PHESS modules in a multi-agent system which is responsible to keep information and profiles about environment, indicators defined and machine learning models updated with sensor information. The dependence on sensor data to create context was noticed and the impact of such workflow was present in terms of network messages between server and sensor nodes, energy efficiency due to active use of sensor nodes and storage size. With the aim to minimize such problem the concept of hybrid virtual sensors was adopted, where in the first stage an intensive use of sensors is performed to learn the behavior of the attributes being sensed through machine learning models and a second stage where these models substitute the sensor data keeping network messages, storage needs and active use of sensors down. Sensor are used at lower frequency rates to assure that the models created remain accurate within a defined error margin. The context server is used to push these models. For instance, one may consider solar exposure where the model takes both location and time to contextualize the number of hours with solar exposure. Such model may be maintained in a context server, not

requiring an active agent. Other examples created within the PHESS project includes photovoltaic panels output according to atmospheric conditions and hour of day and exterior temperature from known weather API's in the internet. Although this approach is able to maintain context scenarios it may lack alert and fast response as the data from sensor is not actively being measured and so should not be used where short term context is relevant but rather historic context.

Middleware access is used to obtain answers to dynamic models maintained for specific environments, users or objects and for direct sensor access. These models developed inside the PHESS project are accessible from external sources through communication APIs developed in JAVA¹, ANDROID² and JADE³ systems which enable the integration of the information created inside this platform available to other initiatives. The communication is made through an ontology written to provide information and data about the environment encapsulating the information displayed by in each API [19]. Such API is used for related projects such as stress assessment, emotional control and gamification purposes.

4 Model Assessment

Over a period of three days a simple application with the PHESS project was evaluated, where sensors for environment luminosity, humidity and temperature were used. It was assessed aspects related to storage space used and reliability of the model created using the theory described in sections 2 and 3. Initial results demonstrated that by adapting the learning rate on attributes monitored, there are gains in terms of storage needed as well as active use of sensors while keeping results with over 70% relative accuracy.

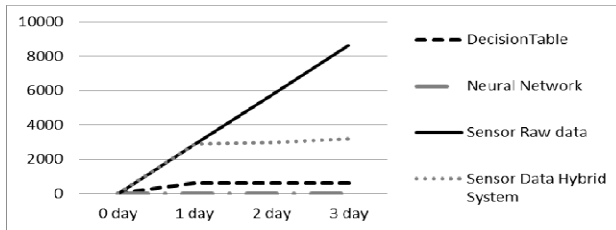


Fig. 3. Space required by each strategy

Figure 3, demonstrates potential savings with this approach, as less storage capacity is used to store sensor data and models being updated generally are fixed in size. The results presented refer to ambient luminosity which was modeled by a decision table algorithm and a multilayer perceptron using room location and time of day. Sensor values were assessed for the case of continuous monitoring (raw data at 1min

¹ Oracle Java. Source: <http://www.java.com/>

² Android Project. Source: <http://www.android.com/>

³ Jade – Java Agent Development Framework. Source: <http://jade.tilab.com/>

intervals) and a hybrid approach with sensor values validating the model created after the learning phase at 30 min. The models created can be used in context server through the PHESS project to simulate context where such models are determinant easing the need to use middleware API to query sensors or databases of historic values.

Table 1, it presented accuracy values for the machine learning models for luminosity, humidity and temperature using a decision table algorithm. These results gather the error which the models are subjected and within which models are not re-trained. The correlation values are the correlation between predicted and real values which was also used to assess models initially.

Table 1. Continous Model Assessment

Model Object	Mean Error	Relative Error	Correlation
Luminosity	102.75	27.04 %	0.88
Temperature	0.46	22.40 %	0.97
Humidity	0.86	23.76 %	0.96

The substitution of models instead of always requiring sensor data reduces message traffic between agents and increases system overall performance. Nevertheless, more precise studies are still necessary in order to maintain the stability of models found and their descriptive soundness. Overall, results are positive, with the approach demonstrated to be both feasible and adequate for the problems being targeted.

5 Conclusion

The work detailed in this paper describes how to take advantage of context awareness by using machine learning models in conjunction with concepts from information fusion inside intelligent systems. This approach, aims to reduce the impact of storage and sensor data problems while maintaining historic behavior description and preserving contextual information about the attribute. The results provided, show promise in maintaining storage levels contained and may be used as a proof-of-concept. The accuracy of models depends on their design, having detailed in this article some sample context models to be applied in these systems. Future development for the models created inside this system encompasses their use for multi-environments. The aim is to also reduce the learning cost of new environments with pre-trained models that are then contextualized in each environment with lifelong learning according to methods similar to ones presented in this paper.

Acknowledgement. This work is part-funded by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project PEst-OE/EEI/UI0752/2011. This work is also part-funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through FCT within project FCOMP-01-0124-FEDER-028980 (PTDC/EEI-SII/1386/2012) and by the doctoral grant SFRH/BD/78713/2011 by FCT.

References

1. Kwak, J., Varakantham, P., Maheswaran, R., Tambe, M.: SAVES: A Sustainable Multi-agent Application to Conserve Building Energy Considering Occupants. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Innovative Applications Track (2012)
2. Klein, L., Kavulya, G., Jazizadeh, F., Kwak, J.: Towards optimization of building energy and occupant comfort using multi-agent simulation. In: Proceedings of the 28th ISARC, pp. 251–256 (2011)
3. Youngblood, G.M., Holder, L.B., Cook, D.J.: Managing Adaptive Versatile Environments. In: Third IEEE International Conference on Pervasive Computing and Communications, PerCom, pp. 351–360 (2005)
4. De Ruyter, B.: Ambient intelligence: visualizing the future. In: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 203–208 (2004)
5. Torra, V.: Information Fusion - Methods and Aggregation Operators. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 999–1008. Springer US (2010)
6. De Paola, A., Gaglio, S., Lo Re, G., Ortolani, M.: Sensor 9 k: A testbed for designing and experimenting with WSN-based ambient intelligence applications. *Pervasive and Mobile Computing* 8(3), 448–466 (2012)
7. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. *International Journal of Ad Hoc...* 2(4) (2007)
8. Nakamura, E.F., Loureiro, A.A.F., Frery, A.C.: Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Comput. Surv.* 39(3) (2007)
9. Khaleghi, B., Khamis, A., Karray, F.O., Razavi, S.N.: Multisensor data fusion: A review of the state-of-the-art. *Information Fusion* 14(1), 28–44 (2013)
10. Xiong, N., Svensson, P.: Multi-sensor management for information fusion: issues and approaches. *Information Fusion* 3, 163–186 (2002)
11. Durrant-Whyte, H., Henderson, T.: Multisensor Data Fusion. In: Siciliano, B., Khatib, O. (eds.) *Springer Handbook of Robotics*, pp. 585–610. Springer, Heidelberg (2008)
12. Wang, K.I.-K., Abdulla, W.H., Salcic, Z.A.: Ambient intelligence platform using multi-agent system and mobile ubiquitous hardware. *Pervasive and Mobile Computing*
13. Bonino, D., Corno, F.: Rule-based intelligence for domotic environments. *Automation in Construction* 19(2), 183–196 (2010)
14. Schilit, B., Adams, N., Want, R.: Context-Aware Computing Applications. In: *First Workshop on Mobile Computing Systems and Applications, WMCSA 1994* (1994)
15. De Ruyter, B.E.R., Aarts, E.: Ambient intelligence: visualizing the future. In: *AVI 2004 Proceedings of the Working Conference on Advanced Visual Interfaces*. ACM (2004)
16. Ortiz, A.M., Royo, F., Galindo, R., Olivares, T.: I3ASensorBed: a testbed for wireless sensor networks (2011)
17. Silva, F., Analide, C., Rosa, L., Felgueiras, G., Pimenta, C.: Ambient Sensorization for the Furtherance of Sustainability. In: van Berlo, A., Hallenborg, K., Rodríguez, J.M.C., Tapia, D.I., Novais, P. (eds.) *Ambient Intelligence & Software & Applications*. AISC, vol. 219, pp. 179–186. Springer, Heidelberg (2013)
18. Silva, F., Analide, C., Rosa, L., Felgueiras, G., Pimenta, C.: Social Networks Gamification for Sustainability Recommendation Systems. In: Omatu, S., Neves, J., Rodriguez, J.M.C., Paz Santana, J.F., Gonzalez, S.R. (eds.) *Distrib. Computing & Artificial Intelligence*. AISC, vol. 217, pp. 307–315. Springer, Heidelberg (2013)
19. Silva, F., Analide, C.: PHESS - People Help Energy Savings and Sustainability - Technical Report (2013)

Simply-Integrated Method of Judgments of Expert Knowledge Collected in Databases for Objective Computer-Aided Engineering Systems

Piotr Michalski, Mariusz Piotr Hetmańczyk, and Jerzy Świder

The Silesian University of Technology, Faculty of Mechanical Engineering,
Institute of Engineering Processes Automation and Integrated Manufacturing Systems,
44-100 Gliwice, Konarskiego 18A Street, Poland
{piotr.michalski, mariusz.hetmanczyk, jerzy.swider}@polsl.pl

Abstract. Prediction of the machinery or process malfunction states is one of the important and challenging task for knowledge engineers. Nowadays it is a common situation, that the reliable companies collect all kind of important knowledge and by the work of knowledge engineers remake it for well known know-how. The problem we would like to solve is touching the area of objective computer-aided engineering systems. The new proposal of the method for judgments of the expert rules collected in the databases which is arbitral in our opinion has been described in this article. New method based on aggregation process of subjective marks of reasoning rules from independent experts, which gives back an more objective parameters for faster computing of the reasoning process. Thanks to this the malfunctions can be reduced to minimum or much faster eliminate.

Keywords: Judgment process, Knowledge acquisition, knowledge discovery, Operator of aggregation.

1 Knowledge Acquisition and Judgment Process

Knowledge acquisition process should be done in such a way that the participation of the knowledge programmer should be reduced to minimum. Usually to prepare the knowledge reasoning system we need to complete it in seven steps [6-8], [11], [12]. Figure 1 shows that the first 4 steps have to be realized by the expert or specialist by themselves (with support of specially prepared paper form or e-form of questionnaires). Then the steps from 5 to 7 have to be realized by the knowledge programmer. All results from every step has to be check by the author of the knowledge database or the other responsible person.

Important task in the process of judgment of the collected knowledge is the aggregation of many subjective marks into one objective parameter. Typically the collected knowledge is judgment by the independent specialist, what is also a method of collecting knowledge [3], [10], [12]. Not only the rules come into the judgment but also the entire knowledge database. The proposed method assume that the data is collected

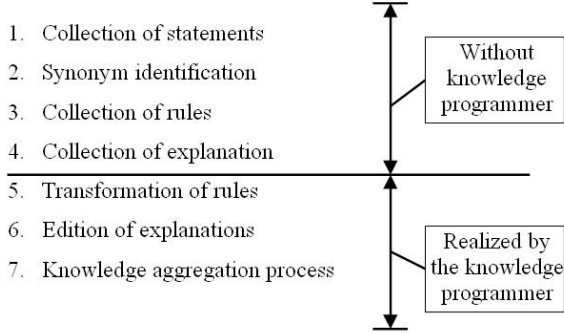


Fig. 1. Stages of the knowledge acquisition process [11]

by the e-forms and stored as a rules for judgment process in the database. In detail we will judge the rules by assign to them the degree of correctness of the rule called $B(r)$.

This new parameter will have value from range $\langle 0,1 \rangle$. Range of variation will be described by two parameters: degree of necessity of the rule $N(r)$, and degree of possibility of the rule $P(r)$, what we can describe as:

$$0 \leq N(r) \leq B(r) \leq P(r) \leq 1 \tag{1}$$

Table 1. Range of variation of the degree of correctness of the rule [11]

Name of a convincing degree of correctness – the answer provided by a particular expert	Necessity of the rule	Possibility of the rule
I absolutely agree	1,00	1,00
I surely agree	0,75	1,00
I rather agree	0,55	1,00
I do not have an opinion	0,00	1,00
I rather disagree	0,00	0,45
I surely I disagree	0,00	0,25
I completely disagree	0,00	0,00

Table 1 shows the set of quality values which are very helpful for specialist or knowledge engineers during the process of establishing the degree of correctness of the rule. We are taking into consideration the fact that the single rule can be judge by many experts, and those judgments can be different, so that is the reason to announce the operators of aggregation. Up to now the best of those was the operator of aggregation of degree of correctness based on value of measures $NP(r)$. In accordance with the rule r :

$$NP_{ag}(r) = \frac{w(r) \cdot NP(r) + w_{ex} \cdot B_{ex}(r)}{w(r) + w_{ex}} \tag{2}$$

where:

- $w(r)$ – importance of the degree of correctness of the rule,
- w_{ex} – importance attributed to expert,
- $B_{ex}(r)$ – value of degree of correctness of the rule assigned by the expert,
- $NP(r)$ – actual value of degree of correctness of the rule by the aggregation of many opinion, calculated as:

$$NP(r) = \frac{N(r) + P(r)}{2} \quad (3)$$

Now the process of judgments of the rules looks like this: the specialist can review all the rules, which have unique id numbers, domain and premises of the rules and conclusions for the rules. He is able to write his own subjective opinion about the every rule (using the prepared degree of correctness of the rule from table 1). Next this opinion is aggregated with the opinion of others experts who judge his rule before.

2 An Existing Operator of Aggregation

Important and very controversy aspect of this process is the assignments of beginning importance attributed to every expert, which has to be done by the knowledge programmer. That is the reason that most common they assign $w(r) = 1$. We have to agree that the operator of aggregation in witch is a place for subjective judgment is not the best solution for future objective computer-aid integrated system. Let's modify it, so we can describe it by the equation:

$$NP_{agos}(r) = \sum_{i=1}^m w(n, k_i) \cdot NP_i(r) \quad (4)$$

where:

- $NP_i(r)$ - degree of correctness of the rule calculated from equation (3), where the value of $N(r)$ and $P(r)$ comes from table 1,
- $w(n, k_i)$ – frequency of occur of the value $NP(r)$, calculated as:

$$w(n, k_i) = \frac{k_i}{n} \quad (5)$$

where:

- k_i – amount of value $NP(r)$,
- n – summary number of opinion about judgment rule.

Value of the parameter we can get after this kind of aggregation process [11] is always in the range of:

$$0 \leq NP_{agos}(r) \leq 1 \quad (6)$$

Thanks to this solution this operator of aggregation can be independent from renown of each expert, and based on opinion of each expert we are able to get the “insertion” or “attenuation” of the judgment – what is relevant to correctness of the rule.

We have to consider the situation of making the wrong judgment – mark high the wrong solution – what is come from the fact that in this operator of aggregation the arithmetic mean answer is the best solution which system will suggest after the reasoning process. So let’s add another restriction to our proposal: lets the system use only those rules which are judged by more then one expert. Those judged by only one expert should not be used in reasoning process, and must be whiting for more opinion from other experts. So now, it is well visible need ([1], [2], [4], [5], [9]) of another modification of the operator of aggregation, which will allow for automation-recognize by the objective system the degree of correctness of the rule taking for the consideration the number of convergent expert opinion.

3 The New Approach

Authors of this article propose the new approach of calculation of the operator of aggregation basing on the geometry of the flat figures (like in fuzzy logic), which can be circumscribe on collection of experts opinion, and we will call it – $NP_{agfp}(r)$. Table 2 shows the calculated values of $NP(r)$ calculated using the equation (3)

Table 2. Collected values of degree of correctness of the rule [8], [12]

Name of a convincing degree of correctness – the answer provided by a particular expert	Necessity of the rule	Possibility of the rule	Collected value $NP(r)$
I absolutely agree	1,00	1,00	1
I surely agree	0,75	1,00	0,875
I rather agree	0,55	1,00	0,775
I do not have an opinion	0,00	1,00	0,5
I rather disagree	0,00	0,45	0,225
I surely I disagree	0,00	0,25	0,125
I completely disagree	0,00	0,00	0

The idea is simple and clever: lets build the flat figure on vertex of segments (vertical position) which represents the sum of identical expert’s answers, and build those segments (horizontal position) in the range [0,1] in position of collected value of $NP(r)$. As a sample let’s consider the situation shown on figure 2, where we have the flat figure circumscribe based on opinion of 10 experts. One marked “I rather disagree”, another “I do not have opinion”, another one “I rather degree”, five of experts marked “I surely agree”, and two others “I absolutely agree”.

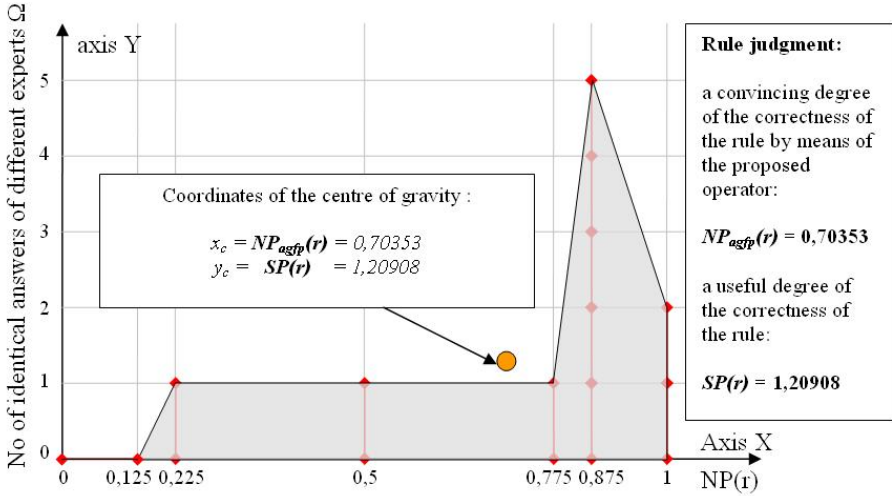


Fig. 2. Sample figure – new operator of aggregation of the expert opinion based on the calculation of the center of gravity of the flat figure

Table 3 shows the X, Y coordinates of the vertex from our sample situation.

Table 3. Vertex coordinates base on which the flat figure was circumscribed

Vertex description	X coordinate	Y coordinate
Start vertex	0	0
I absolutely agree	0	0
I surely agree	0,125	0
I rather agree	0,225	1
I do not have an opinion	0,5	1
I rather disagree	0,775	1
I surely I disagree	0,875	5
I completely disagree	1	2
End vertex	1	0

After calculation of the center of gravity of this flat figure by using the equations (7) and (8) we get convincing degree of correctness of the rules $NP_{agfp}(r)$ (which we assign to X coordinate – x_c) equal to: 0,70353, and the other new parameter called degree of usability of the rule $SP(r)$ (assign to Y coordinate – y_c) equal to: 1,20908. Using this new parameter $SP(r)$ we are able to proceed with automation of the choosing the rules for the reasoning process – simple for computer implementation. Just for remind how to calculate the centre of gravity of the flat figure we put useful equations (7) and (8) bellow.

$$x_c = NP_{agfp}(r) = \frac{\sum_{i=1}^n A_i x_i}{\sum_{i=1}^n A_i} \quad (7)$$

$$y_c = SP(r) = \frac{\sum_{i=1}^n A_i y_i}{\sum_{i=1}^n A_i} \quad (8)$$

Just for comparison, the convincing degree of correctness of the rule calculated with old operator of aggregation (4), based on the same input values looks like: $NP_{agos}(r) = 0,7875$.

Value of the parameter $NP_{agfp}(r)$ we can get after usage of the new operator (7, 8) of aggregation process is always in the range of:

$$0,04 \leq NP_{agfp}(r) \leq 0,96 \quad (9)$$

Typically the existing expert systems use so called *Certainty Factor* – CF , so to be able to assign values of our new shown operator of aggregation $NP_{agfp}(r)$ to the CF , which values are in range of $[0,1]$, where 0 means false rule, and 1 means true rule, we have to change the edge conditions, by closing it both side like that:

$$NP_{arfp} \Rightarrow CF \begin{cases} NP_{arfp} \leq 0,04 \rightarrow CF = 0 \\ 0,041 \leq NP_{arfp} \leq 0,95 \rightarrow CF = NP_{arfp} \\ NP_{arfp} \geq 0,96 \rightarrow CF = 1 \end{cases} \quad (10)$$

We can notice, that rules which are judge positive by many experts the final mark is relatively high. Taking into consideration that fact we can certify that the reasoning process will proceed correct. The low marks can mean deficiency correctness of the rules or even more – occur new rule which is in total opposition from well known knowledge. This total opposition could be an exception, which has to be added to the database of knowledge.

Degree of usability of the rule $SP(r)$ reach the value equal to 1 after minimum 3 the same experts answers (opinion). Next opinion will increase or decrease this parameter depends on positive or negative opinion of next expert about considered rule. Now the engineer of knowledge (not programmer of knowledge) can setup the trigger point level value, which will admit the rule to use in reasoning process between the other active one. System should allowed to change opinion about the rule judged by the expert before. After that kind of situation should one again calculate the $SP(r)$ parameter. New expert in the system, and his opinion should be also the reason for that behavior.

4 Conclusions

The problem we solve is touching the area of objective computer-aided engineering systems. The new proposal of the method for judgments of the expert rules collected

in the databases which is arbitral in our opinion has been reached by the introducing the new method based on aggregation process of subjective marks of reasoning rules from independent experts, which gives back an more objective parameters for faster computing of the reasoning process. Operator of aggregation $NP_{agfp}(r)$ comes from the calculation of the center of gravity of the flat figure circumscribed in special way on the vertex of segments which represents the sum of identical expert's answers. Additionally after this new calculation we get new parameter called degree of usability of the rule $SP(r)$ which will help to create the objective reasoning system, independent from assign the importance attribute to every expert. Thanks to this kind of new objective reasoning systems the malfunctions of any system can be reduced and critical states much faster eliminate.

References

1. CEDES AG. Safety and automation. Landquart (2005) ISBN 3-9522402-1-4
2. Dolderer, P., Teichmann, R.: EMC The easy way - Pocket Guide. ZVEI, Frakfurt (2004)
3. Hetmanczyk, M.P., Michalski, P., Swider, J.: Utilization of advanced self-diagnostic functions implemented in frequency inverters for the purpose of the computer-aided identification of operating conditions. *J. Vibroeng.* 14(1), 117–122 (2012)
4. Kriesel, W.R., Madelung, O.W., Werner, R.: AS_Interface the Actuator-Sensor Interface for Automation. Hanser, Wien (1999)
5. Kriesel, W.R., Madelung, O.W., Werner, R.: AS_Interface The Actuator-Sensor Interface for Automation. Supplement to the 2nd edn. Hanser, Wien (2002)
6. Krenczyk, D., Kalinowski, K., Grabowik, C.: Integration Production Planning and Scheduling Systems for Determination of Transitional Phases in Repetitive Production. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012, Part II. LNCS, vol. 7209, pp. 274–283. Springer, Heidelberg (2012)
7. Grabowik, C., Krenczyk, D., Kalinowski, K.: The Hybrid Method of Knowledge Representation in a CAPP Knowledge Based System. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012, Part II. LNCS, vol. 7209, pp. 284–295. Springer, Heidelberg (2012)
8. Michalski, P., Swider, J.: System wspomagania diagnostyki sieci przemysłowych AS-interface, Computer-aid system for AS-interface industrial network diagnosis. WPSI, Gliwice (2008) (in Polish) ISBN 978-83-7335-478-4
9. Madelung, O.: AS-Interface in Real-Time Environment. *Real-Time Magazine* (4), 88–90 (1997)
10. Swider, J., Michalski, P., Wszolek, G.: Physical and geometrical data acquiring system for vibration analysis software. *Journal of Materials Processing Technology* 164, 1444–1451 (2005)
11. Moczulski, W.A.: Metody pozyskiwania wiedzy dla potrzeb diagnostyki maszyn, Methods of the data acquisition for machines diagnosis process. *ZN Pol. Śląskiej* nr 1382, s. Mechanika, z. 130, Gliwice (1997) (in Polish)
12. Moczulski, W.A.: Diagnostyka techniczna. Metody pozyskiwania wiedzy. *Diagnostics. Knowledge acquisition methods. Monographic, Politechnika Śląska, WPSI, Gliwice* (2002) (in Polish)

A Hybrid Inference Approach for Building Fuzzy DSSs Based on Clinical Guidelines

Aniello Minutolo, Massimo Esposito, and Giuseppe De Pietro

Institute for High Performance Computing and Networking, ICAR-CNR
Via P. Castellino, 111-80131, Napoli, Italy

Abstract. Clinical practice guidelines are expected to promote more consistent, effective, and efficient medical practices, especially if implemented in clinical Decision Support Systems (DSSs). With the goal of properly representing and efficiently handling clinical guidelines affected by uncertainty and interconnected between them, this paper proposes a hybrid fuzzy inference approach for building fuzzy DSSs. It provides a set of specifically devised functionalities for best modeling and reasoning on the particular clinical knowledge underpinning guidelines: i) it organizes the whole fuzzy DSS into self-contained sub-systems which are able to independently reason on piece of knowledge according to their peculiar inference scheme; ii) a global inference scheme has been defined for handling and reasoning on such sub-systems, according to the classical crisp expert system approach. As a proof of concept, the proposed approach has been applied to a practical case, showing its capability of supporting multiple levels of inference and, thus, highlighting the possibility of being profitably used to model and reason on complex clinical guidelines in actual medical scenarios.

Keywords: Decision Support Systems, Fuzzy Logic, Clinical Guidelines, Multi-Level Inference.

1 Introduction

Clinical Decision Support Systems (DSSs) are information systems having highly sophisticated reasoning capabilities designed not to supplant, but to improve clinical decision making, and, thus, promote the efficiency of medical practices. Such systems have been evaluated as having the most efficient impact when they embed a computer-interpretable formalization of clinical guidelines and provide patient-specific advice at the time and place of a consultation. Such a way, clinicians can have access to not just the decision, but also the set of clinical guidelines from which it was derived, and the literature that explains their scientific evidence.

Clinical guidelines are standard specifications of diagnostic or therapeutic recommendations containing the best scientific evidence and experts' opinions [1]. Most clinical guidelines specify isolated care recommendations, describing the dependencies between one or more conditions to satisfy, and the outcomes regarding diagnosis or therapy. Such conditions are usually referred to a patient state, which

describes his/her clinical situation in terms of physiological parameters, symptoms, and execution stages of therapies or medications [2].

Since clinical guidelines are often pervaded by uncertainty and vagueness in their recommendations, in the past Fuzzy logic has been profitably used for representing clinical guidelines in the form of transparent and linguistic conditional statements [3-5]. Such linguistic statements are typically represented in the form of "if-then" rules, named fuzzy rules, by providing an understandable language for describing clinical guidelines in a natural manner close to the human perception.

A drawback of the existing approaches for formalizing clinical guidelines in terms of fuzzy rules within a clinical DSS relies on the absence of any type of vertical arrangement for the specific type of knowledge modeled. Indeed, first of all, the most part of existing clinical DSSs are not able to reason in a single step on rules linked between them, i.e. one or more rules with an antecedent condition which matches an assertion in the consequent of one or more other rules. In other words, they do not address the rule chaining in the inference process and, thus, are only able to work preliminary with the first level of rules, then with the second one and so on. Obviously, such a kind of inference approach does not appropriately fit the peculiarities of the structure of clinical knowledge. Indeed, in order to compute its outcome, e.g. an adjustment of a therapy, a clinical guideline may require output values produced by other guidelines, e.g. the new health status of the patient depending on a worsening of its conditions. Thus, a general-purpose inference approach without any form of flexible support for the rule chaining can rapidly lead to an impoverishment of the level of transparency and maintainability of the whole rule base. Indeed, guidelines linked between them could be reformulated in order to eliminate such a dependence and be processed according to a classical inference approach but implying an exploding number of rules to be handled.

Second, typically a clinical guideline can be formalized as a set of one or more fuzzy rules, which shares the same outcome [6]. For instance, a guideline for expressing the diagnosis of a disease in terms of its stage can be encoded by different rules depending on the number of possible stages. As a consequence of that, it is thinkable that rules encoding the same guideline should also share the same inference configuration for producing their outputs. However, typically, the fuzzy rules are inserted all together in the rule base, even if they encode different guidelines, and, for this reason, are evaluated by exploiting the same inference configuration without offering the possibility of grouping rules depending on the guideline they model and, thus, characterizing them with a specific inference arrangement.

Finally, guidelines are typically distilled in terms of incomplete rules, i.e. they do not cover all possible outputs but just express only pieces of positive evidence, e.g. for expressing the presence of a disease. In this respect, existing solutions do not provide any facility for modeling the negative evidence pertaining a guideline, e.g. the absence of a disease, thus forcing clinicians to formalize a complementary set of rules also for modeling this situation.

Starting from the above considerations, this paper proposes a hybrid inference approach expressly thought for building clinical DSSs, i.e. for efficiently modeling and handling the peculiarities of clinical knowledge encoded in the form of guidelines. In particular, its main features are the following: i) the whole DSS embedding clinical guidelines is organized into sub-systems, each one modeling a

specific guideline; ii) a global hybrid inference scheme is defined to reason on all the sub-systems conceived as a whole and efficiently enable their chaining by exploiting their outcome; iii) each sub-system offers a set of facilities for modeling and reasoning on the care recommendations pertaining a single guideline, by implementing a customized and peculiar inference scheme.

The remainder of the paper is structured as follows. Section 2 reports some considerations about background and related work. Section 3 provides a detailed description of the proposed hybrid inference approach. Section 4 presents a proof of concept scenario to show the application of the proposed solution. Finally, section 5 concludes the work.

2 Background and Design Considerations

Fuzzy logic provides a suitable approach for expressing and formalizing human reasoning in the presence of uncertainty [7], by using an intuitive language close to the human perception. It is based on the notion of *linguistic variable*, such as *child age*, which can assume a determined set of *linguistic values*, such as *low*, *average*, and *high*. Each *linguistic value* is associated with a *fuzzy set*, i.e. an extension of the classical set, whose elements belong to the set with a specific degree of uncertainty defined through a *membership function*, defined in $[0,1]$. The range of numerical values a linguistic variable can assume is called *universe of discourse*.

After modeling linguistic variables, a Fuzzy Inference System (FIS) can be defined to specify a decision-making procedure in terms of linguistic sentences over fuzzy variables, expressed in terms of if-then rules. In this way, in a clinical fuzzy DSS, guidelines can be formalized by describing the dependencies between the symptoms and/or patients' sensing data, and the outcomes of the diagnosis in an understandable way, aiding physicians to understand the decision process followed by the system.

Existing fuzzy DSSs typically use a general purpose inference approach where all fuzzy rules are organized into a whole rule base containing all the modeled clinical guidelines. Such an approach, in complex medical settings, could produce a huge rule base, hard to maintain and inspect when many group of rules need to be connected between them for sharing their outcomes. In order to improve the maintainability of complex rule bases, two main approaches have been proposed in literature.

The first one [8] is based on a hierarchal architecture where the whole FIS is organized in reduced sub-systems which are properly interfaced and connected between them for composing the final output of the system. This approach let to improve the transparency and interpretation of the whole system, since every sub-system represents a self-contained FIS responsible for making inference only on a part of the whole rule base, and the whole FIS's outputs are produced through the propagation of inferred data through the designed hierarchical connections.

The second approach [9, 10] is based on classical expert systems' architectures, where the interconnections between rules do not influence the architecture of the whole system, and the propagation of inferred data is assured by a dedicated component, called *inference engine*, which is responsible for continuously comparing known data with rules for determining and executing the eligible ones.

On the one hand, the hierarchical approach let to organize a clinical DSS through different self-contained inference stages, with different granularity levels, each one modeling a particular guideline and implementing a specific configuration for the inference, but it usually presents a low level of upgradability because of the strict relation between the system architecture and the modeled knowledge. On the other hand, the expert system approach let to easily update the modeled knowledge, but it does not usually provide any form of facility for improving the transparency and interpretation of the system in case of complex clinical DSSs.

The approach proposed in this work has been conceived and developed in order to address these limitations, as described in the following sections.

3 The Hybrid Fuzzy Inference Approach

In accordance with the growing awareness that the combinations of intelligent techniques frequently perform better than the individual ones [11, 12], this paper proposes a hybrid fuzzy inference approach which combines the main advantages of the two afore-mentioned approaches by providing a set of specifically devised functionalities for best modeling and reasoning on the particular clinical knowledge underpinning guidelines. In particular, the approach organizes the whole fuzzy DSS into self-contained sub-FISs which are able to independently reason according to their peculiar inference scheme. Such a way, each clinical guideline can be modeled through a specific sub-system thus improving the level of maintainability of the whole system and enabling the customization of the inference parameters used for each guideline. On the other hand, in order to easily propagate inferred values between sub-FISs encoding single guidelines, without establishing a strict relation between them and the system architecture, the global inference scheme used for handling and reasoning on such sub-systems, has been designed according to the classical crisp expert system approach. In the following more details will be given about how clinical guidelines are formalized and both the global and local inference schemes adopted for reasoning on them.

3.1 Structuring Clinical Guidelines into a Fuzzy DSS

According to the proposed approach, medical knowledge contained in the set of guidelines of interest and embedded into a clinical fuzzy DSS is organized by combining two types of knowledge, namely *declarative knowledge* and *procedural knowledge*. On the one hand, declarative knowledge includes all the fuzzy linguistic variables mentioned in the modeled clinical guidelines as well as their structural parameters, such as name, universe of discourse, and the composing fuzzy sets. On the other hand, procedural knowledge is represented by all the fuzzy rules built on top of the fuzzy linguistic variables and applied by the system to determine its behavior. These two kinds of knowledge form the *Knowledge Base (KB)* and the *Rule Base (RB)* of a fuzzy DSS, respectively.

In order to improve the level of overall maintainability, the whole DSS is organized in different self-contained sub-FISs, each on them being in charge of handling and reasoning on a single specific piece of knowledge embedded in it. In particular, the KB is shared between the sub-FISs, whereas the RB is partitioned depending on its procedural knowledge, i.e. different groups of rules are identified, each of them encoding one single guideline. Every group of rules, which contains the same consequent variable and typically shares the same inference configuration, is configured as a self-contained piece of knowledge to be assigned to a sub-FIS. Indeed, it allows generating the outcome formulated in the guideline independently of the other group of rules, by only working on its own rules and on the linguistic variables included in them. Rules belonging to a group are characterized as belonging to two different typologies, namely rules for expressing the positive evidence of a situation or condition and ELSE rules [13] for expressing the complementary negative evidence. For each group, one or more positive evidence rules are admitted, whereas a single ELSE rule can be defined at most. In order to improve the level of transparency and maintainability offered by a fuzzy DSS and, thus, grant its efficient application in healthcare settings, KB and RB are systematically expressed through an XML-based language [14] describing all the involved linguistic variables, the group of rules built on top of them, and the inference parameters to use for them.

Finally, since the Mamdani-type inference is usually preferred in fuzzy DSS due to the interpretable and intuitive nature of fuzzy sets, the proposed approach has been mainly focused on the aim of supporting the propagation of conclusions in their fuzzy form so as to preserve the complete information produced by sub-FISs. In this respect, a *Working Memory* (WM) is adopted to memorize the specific fuzzy values assumed by each linguistic variable. Fuzzy values are interpreted as fuzzy evidences (known as *fuzzy facts*) on the corresponding fuzzy variable, i.e. a peculiar fuzzy set defined over the universe of discourse associated to a linguistic variable.

Fuzzy facts stored into the WM are updated either in case of new external input data submitted to the whole FIS, or in case of new knowledge inferred through a sub-FIS. In this respect, on the one hand, when input data are submitted in the form of crisp data values associated to some variables in the KB, such crisp data values have to be transformed into fuzzy values, through a *Fuzzifier* component, for coherently storing them into the WM. On the other hand, when required, a *Defuzzifier* component can be invoked for processing fuzzy evidences and associating them a defuzzified value, i.e. a numeric value.

3.2 The Global Inference Scheme

A sub-FIS constitutes a self-contained inference system associated to a single clinical guideline and, thus, it contains all the required information to make inference on the linguistic variable modeling the clinical guideline's outcome. Each sub-FIS applies its own inference scheme independently from the other existing sub-FISs, and demands to the global inference scheme the proper propagation of its conclusions through the WM. The global inference scheme is aimed at combining and propagating the fuzzy knowledge inferred through the different sub-FISs so as to make the DSS able to perform the decision-making process as whole and produce the final outcome.

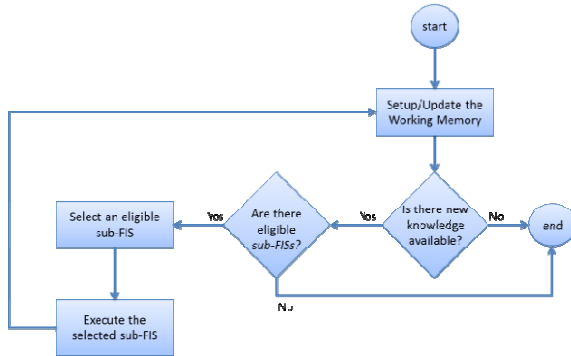


Fig. 1. The global inference scheme used in the proposed hybrid approach

In more detail, it operates as indicated in Figure 1. At the beginning of the inference process, the RB is evaluated for generating the required sub-FISs in accordance with the specified groups of rules and the inference configuration to use. Successively, in accordance with the inference scheme typically used in expert systems, the global inference process is led by the evaluation of all known facts stored in the WM with the goal of recognizing all the eligible sub-FISs, i.e. all the sub-FISs whose RBs contain rules with one or more linguistic variables contained in their antecedent parts for which new fuzzy facts are available. When some eligible sub-FISs are available, the global inference process is paused, one of the eligible sub-FISs is selected and its execution is demanded to a specific component in charge of applying the peculiar inference scheme requested by the selected sub-FIS.

After the execution of the local inference scheme associated to the selected sub-FIS, the global inference scheme will be resumed and it will be repeated until no eligible sub-FIS is available and no further new knowledge is inferred by them.

3.3 The Local Inference Scheme

The local inference scheme defined to operate within a sub-FIS has been designed in accordance with the *First Infer Then Aggregate* paradigm as reported in Figure 2. In this respect, it is important to note that, since the sub-FISs contain all the existing rules able to reason on a linguistic variable, their conclusions are used to eventually update the WM without violating the dictates of Fuzzy Logic in case of multiple levels of inference [10], i.e. an assertion on a fuzzy variable cannot be used for triggering another rule until all the existing outputs entailed by such an assertion have been computed. In detail, the set of positive evidence rules plus the eventual ELSE rule belonging to a sub-FIS is treated as a disjunctive one, i.e. each rule is evaluated independently of other ones, and, at the end, rules' contributions are aggregated for composing the final conclusion of the sub-FIS. In particular, the conclusion inferred by each positive evidence rule is computed by evaluating the linguistic variables involved in the rule's antecedent part against their current values stored in the WM for determining the *strength* of the rule, i.e. the degree of match of its antecedent part.

On the other hand, the outcome of the ELSE rule is generated starting from the strength of all the positive evidence rules in the sub-FIS, since the strength of the ELSE rule is as higher as the strength of other rules is lower. In this way, in case of chained rules inside a sub-FIS, the conclusion inferred by one of them will not be able to immediately alter the inference outcome of other rules. At the end of the execution of a sub-FIS, its final output is evaluated for updating the WM. When new knowledge has been produced, i.e. the shape of the fuzzy outcome produced is significantly different from the last one stored, the sub-FIS is evaluated for determining if it has to be marked to be eligible again. In this case, the inference process on the selected sub-FIS will be restarted. Otherwise, the local inference process will be executed on another sub-FIS until all the eligible sub-FISs will be processed.

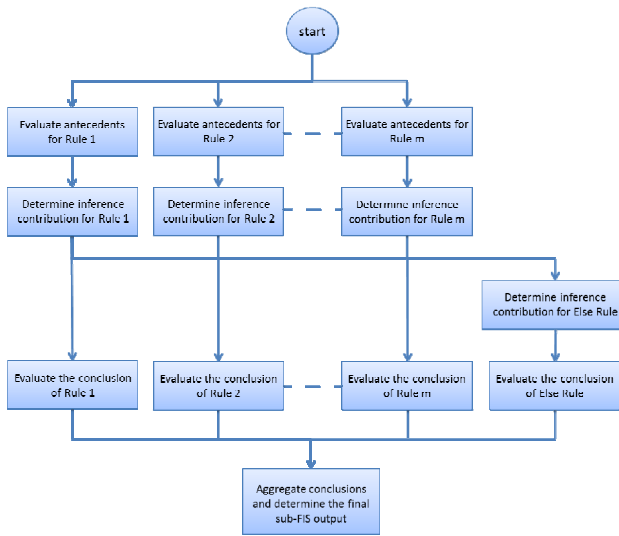


Fig. 2. The local inference scheme used inside a sub-FIS

Furthermore, with the goal of guarantee the propagation of knowledge throughout the sub-FISs, when the execution of a sub-FIS is terminated, its output is evaluated for be eventually used to update the known fuzzy facts previously stored in the WM.

In more detail, when a new fuzzy fact is asserted, the previous value associated to the linguistic variable has to be evaluated in order to determine if the inferred value represents new information to store into the WM. If the previous value does not exist, a new fuzzy fact is directly created. Otherwise, the previous fuzzy fact and the new inferred one have to be compared in order to determine the degree of similarity between them. Two fuzzy facts are considered as similar as their membership functions assign approximately the same values of membership to the elements in their universe of discourse [15]. Afterwards, the fuzzy fact stored into the WM will be updated only if their similarity is smaller than a determined threshold.

After possibly updating the WM with the produced knowledge, the execution of the selected sub-FIS is terminated and the global inference scheme is resumed.

4 A Practical Case of Application

In this section, in order to verify the proposed approach in a practical medical case, some clinical guidelines extracted from the GOLD guideline [16] have been modeled into a fuzzy DSS. GOLD guideline reports the best evidence for diagnosing and managing the Chronic Obstructive Pulmonary Disease (COPD), formalized by the US National Heart, Lung, and Blood Institute and the World Health Organization in 2006. COPD is characterized by the presence of significant airflow obstruction and some extra-pulmonary effects that can lead to frequent hospitalization and eventually death from suffocation. COPD diagnosis is performed through the spirometry test which establishes the presence of airflow obstruction, by measuring the forced expiratory volume in one second (FEV_1) and the forced vital capacity ratio (FEV_1/FVC). Starting from the spirometry test, and in addition to the presence of Chronic Respiratory Failure (CRF), the GOLD guideline reports a set of clinical rules for diagnosing the COPD and classifying its severity, as shown in Table 1.

Table 1. The GOLD guideline’s recommendations for the COPD diagnosis

FEV ₁ / FVC ratio	FEV ₁ predicted	Presence of CRF	COPD severity
< 0.70	≥ 80%	-	Mild
< 0.70	within [50%, 80%[-	Moderate
< 0.70	within [30%, 50%[-	Severe
< 0.70	< 30 %	-	Very Severe
< 0.70	< 50 %	Yes	Very Severe

Moreover, in accordance with the medical progress and the evolution in the COPD severity, the GOLD guideline suggests the proper medical treatment to follow, as shown in Table 2. For instance, while the use of influence vaccines is recommended for COPD patients over 65 years and older, the use of inhaled glucocorticosteroids is suggested only for COPD patients who have manifested repeated exacerbations, i.e. further amplification of the inflammatory response in their airways.

Table 2. The GOLD guideline’s recommendations for the COPD treatment

COPD severity	COPD exacerbations	Presence of CRF	Patient age	COPD treatment
Mild, Moderate, Severe, Very Severe	-	-	≥ 65	Influenza vaccines
Very Severe	-	-	< 65	Influenza vaccines
Mild	-	-	-	Short-acting bronchodilators
Moderate, Severe, Very Severe	-	-	-	Long-acting bronchodilators
Severe or Very Severe	Repeated	-	-	Inhaled glucocorticosteroids
Very Severe	-	Yes	-	Long term oxygen

The formalization of these guidelines in Fuzzy Logic generates two groups of fuzzy rules, one for each guideline, where the knowledge inferred by the first group is evaluated by the second one for determining the proper medical treatment according to the current stage of the COPD severity.

Differently from existing solutions [3-5], where all fuzzy rules modeling the clinical guidelines are arranged into a FIS based on a unique rule base, in the proposed approach the two groups of rules can be arranged to form two self-contained systems, the *COPD severity sub-FIS* and the *COPD treatment sub-FIS*. Each sub-FIS has been constructed by specifying rules, linguistic variables involved in them and the

inference parameters to be used. In particular, all the features represented as columns in the two tables reported above have been formalized as linguistic variables of the final fuzzy DSS. Moreover, the clinical rules reported as rows in the two tables have been fuzzified and encoded as fuzzy rules.

Thus, after constructing in such a way the two sub-FISs, the fuzzy DSS encoding the afore-mentioned guidelines can be used. For instance, given a 70-years old patient having a value of 70% for the FEV₁ predicted, and a value of 60% for the FEV₁/FVC ratio, with neither manifestations of CRF nor exacerbations, five fuzzy assertions are generated, through the Fuzzifier, and stored into the WM.

Since no fuzzy assertion exists, which involves the *COPD severity* variable contained in all the antecedent parts of all the rules in the *COPD treatment sub-FIS*, the global inference scheme selects only the *COPD severity sub-FIS* as eligible and, thus, enables it to be executed. In accordance with the facts stored into the WM, all rules in the *COPD severity sub-FIS* are independently computed for determining their activation degrees, so as to compose the contribution of each rule to the final conclusion of that sub-FIS. In this respect, Figure 3 reports the activation degrees computed for the rules, and the inferred fuzzy evidence for the *COPD severity* and, afterwards, for the *COPD treatment*.

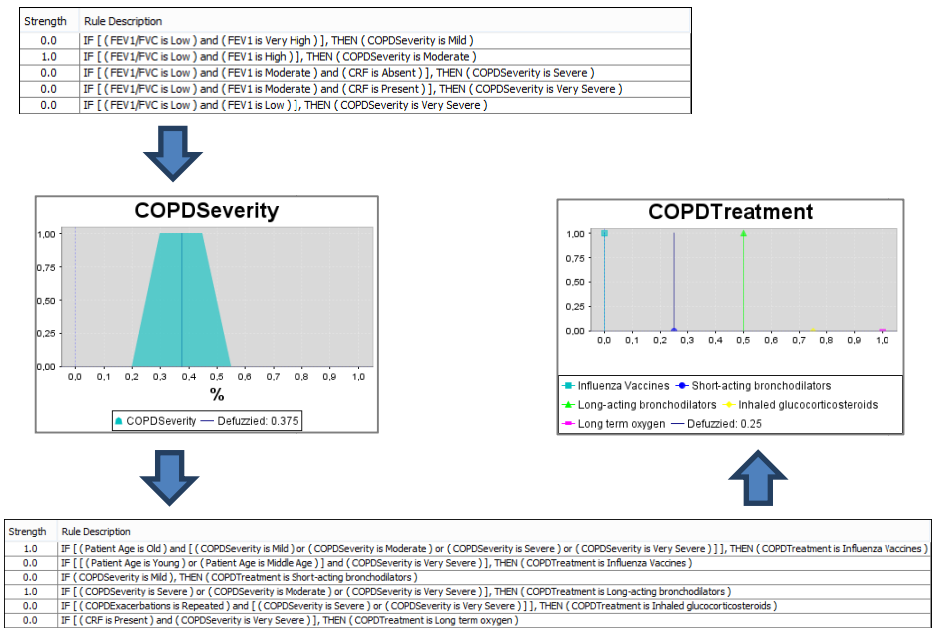


Fig. 3. The generated fuzzy facts for *COPD severity* and, afterwards, for *COPD treatment*

Since the new inferred knowledge does not impact the premises of the considered sub-FIS, it does not require any further execution and the global inference scheme can be resumed. At this point, the *COPD treatment sub-FIS* has become eligible and is selected for the execution. Thus, the previous inferred fuzzy fact is evaluated and a new fuzzy evidence for *COPD treatment* is generated. It is worth noting that the

evaluation of rules of the *COPD treatment sub-FIS* against the generated fuzzy fact for the *COPD severity* is performed by determining the overlapping area between the conditions in the premises and that fuzzy fact. In this case, the degree of match of a condition is determined by selecting, in such an area, the highest degree of membership to that condition. Finally, since no other sub-FIS is eligible, and, no further fuzzy evidence is generated, the global inference process can be terminated.

Differently from the exiting solutions, the choice of confining rules in sub-systems, each one modeling a specific clinical guideline, let to increase the transparency of the whole system's traceability enabling a facilitated characterization of the execution flow in terms of clinical guidelines, i.e. reporting which ones have been activated from fuzzy evidences and/or which ones are responsible of the final system's outcomes (see Fig. 3). Moreover, since the WM has been designed for managing inferred facts in their fuzzy form, the sub-FISs' conclusions are propagated by preserving their complete information which, instead, would be lost if propagated through their defuzzified values as existing clinical fuzzy DSSs typically do.

However, it has to be highlighted that a potential drawback of the proposed approach could be the way rules are confined for composing sub-systems and modeling clinical guidelines. On the one hand, this characteristic can be very useful in case of clinical guidelines expressed in form of fuzzy rules containing the same linguistic variable in their consequent part. On the other hand, in case of general domain modeling and/or when different rule structures are involved, this behavior could not be desired and/or become counterproductive. In this respect, if a complex clinical guideline is potentially described as composed by different fuzzy rules acting on distinct linguistic variables, the proposed approach forces the partitioning of such guideline in more sub-guidelines/sub-systems increasing de facto the granularity level required to model clinical guidelines into the whole fuzzy DSS.

5 Conclusions

This paper presented a hybrid fuzzy inference approach for building fuzzy DSSs which provides a set of specifically devised functionalities for best modeling and reasoning on the particular clinical knowledge underpinning guidelines. In particular, the approach organizes the whole fuzzy DSS into self-contained sub-FISs which are able to independently reason according to their peculiar inference scheme. Such a way, each clinical guideline can be modeled as a group of rules embedded into a specific sub-system, thus enabling the customization of the inference parameters used for each guideline. On the other hand, a global inference scheme has been defined for handling and reasoning on such sub-systems, according to the classical crisp expert system approach.

Differently from exiting fuzzy DSS based on clinical guidelines, the strength of this hybrid approach relies on its capability of enabling the partitioning of complex fuzzy DSS into different sub-systems, with different granularly levels, which are able to share and exchange fuzzy knowledge without establishing a strict relation between the system architecture and the way inferred values have to be propagated through them. The proposed approach let to increase the transparency of the whole system's traceability enabling both the characterization of the execution flow in terms of

clinical guidelines (the modeled sub-systems), and the proper propagation of their conclusions since the WM is designed for managing facts in their fuzzy form.

Finally, as a proof of concept, the proposed approach has been applied to a practical case, showing its capability of supporting multiple levels of inference and, thus, highlighting the possibility of being profitably used to model and reason on complex clinical guidelines in actual medical scenarios.

References

1. Leape, L.: Practice guidelines and standards: An overview. *QRB, Quality Review Bulletin* 16(2), 42 (1990)
2. Wang, D., Peleg, M., Tu, S.W., Boxwala, A.A., Greenes, R.A., Patel, V.L., Shortliffe, E.H.: Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: A literature review of guideline representation models. *Int. Journal Med. Inform.* 68(1-3), 59–70 (2002)
3. Ainon, R.N., Bulgiba, A.M., Lahsasna, A.: AMI Screening Using Linguistic Fuzzy Rules. *Journal of Medical Systems* 36(2), 463–473 (2012)
4. Adeli, A., Neshat, M.: A fuzzy expert system for heart disease diagnosis. In: *Proc. of International Multiconference of Engineering and Computer Scientists*, pp. 134–139 (2010)
5. Lahsasna, A., Ainon, R.N., Zainuddin, R., Bulgiba, A.: Design of a Fuzzy-based Decision Support System for Coronary Heart Disease Diagnosis. *JM Syst.* 36, 3293–3306 (2012)
6. Shiffman, R.: Representation of clinical practice guidelines in conventional and augmented decision tables. *J. of the American Medical Informatics Association* 4(5), 382–393 (1997)
7. Zadeh, L.A.: A theory of approximate reasoning. In: *Machine Intelligence*, pp. 149–194. John Wiley & Sons, New York (1979)
8. Torra, V.: A review of the construction of hierarchical fuzzy systems. *Int. J. Intell. Syst.* 17, 531–543 (2002)
9. Sottara, D., Mello, P., Proctor, M.: Adding Uncertainty to a Rete-OO Inference Engine. In: *Proc. of the Int. Symposium on Rule Representation, Interchange and Reasoning on the Web*, pp. 104–118 (October 2008)
10. Pan, J., Desouza, G.N., Kak, A.C.: Fuzzyshell: a large-scale expert system shell using fuzzy logic for uncertainty reasoning. *IEEE Trans. Fuzzy Syst.* 6, 563–581 (1998)
11. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. *Neurocomputing* 75(1), 61–63 (2012)
12. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. *Information Sciences* 180(14), 2633–2634 (2010)
13. Esposito, M., De Falco, I., De Pietro, G.: An evolutionary-fuzzy DSS for assessing health status in multiple sclerosis disease. *Int. J. of Med. Inf.* 80(12), e245–e254 (2011)
14. Minutolo, A., Esposito, M., De Pietro, G.: A Fuzzy Decision Support Language for building Mobile DSSs for Healthcare Applications. In: Godara, B., Nikita, K.S. (eds.) *MobiHealth 2012. LNCS*, vol. 61, pp. 263–270. Springer, Heidelberg (2013)
15. Setnes, M., Babuska, R., Kaymak, U., van Nauta Lemke, H.: Similarity measures in fuzzy rule base simplification. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics* 28(3), 376–386 (1998)
16. Rabe, K., Hurd, S., Anzueto, A., Barnes, P., Buist, S., Calverley, P., Fukuchi, Y., Jenkins, C., Rodriguez-Roisin, R., van Weel, C., et al.: Global strategy for the diagnosis, management, and prevention of chronic obstructive pulmonary disease: Gold executive summary. *American Journal of Respiratory and Critical Care Medicine* 176(6), 532 (2007)

Hybrid Visualization for Deep Insight into Knowledge Retention in Firms

Lourdes Sáiz¹, Miguel A. Manzanedo¹, Arturo Pérez²,
Álvaro Herrero¹, and Emilio Corchado³

¹ Department of Civil Engineering, University of Burgos
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain

² Investigador Programa Torres Quevedo, TTT Diseño, Comunicación y Contenidos, S.L.
Reyes Católicos, 41, 1, 09005 Burgos, Spain

³ Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced, s/n, 37008 Salamanca, Spain
{lsaiz,mmanz,ahcosio}@ubu.es, arrturo@hotmail.com,
escorchado@usal.es

Abstract. Neural projection models are applied in this study to the analysis of Human Resources (HR) from a Knowledge Management (KM) standpoint. More precisely, data projections are combined with the glyph metaphor to analyse KM data and to gain deeper insight into patterns of knowledge retention. Following a preliminary study, the retention of specialized employees in hi-tech companies is investigated, by applying the configurational approach of Strategic HR Management. The combination of these two aforementioned techniques generates meaningful conclusions and the proposal is validated by means of an empirical study on a real case study related to the Spanish hi-tech sector.

Keywords: Unsupervised Neural Networks, Glyph Metaphor, Knowledge Management, Knowledge Retention, Human Resources Management.

1 Introduction

Knowledge Management (KM) [1] means that organizations can capture and share the collective experience and the know-how (knowledge) of their employees and apply their knowledge in intelligent ways [2]. However, before an organization can successfully apply a KM methodology, it first has to develop and to implement its knowledge infrastructure [3]. These knowledge infrastructures consist of three central dimensions: people, organizational and technological systems.

It is no exaggeration to say that in an environment such as today's, where everything changes at great speed and almost nothing remains static, knowledge emerges as the key factor in any economy [4]. Knowledge not only means that we can advance, but that we can adapt to the unceasing change that happens around us, which are bound to increase in the future. Equally, the firm inevitably requires "specific/singular" knowledge, which will permit it to pursue excellence alongside

others. This class of first-level knowledge is held by a small number of people, without forgetting that more select and sophisticated knowledge is required, if possible, in pioneering firms, such as those analyzed in this study, in an area where the offer is, unfortunately, further and further away from satisfying demand.

Here, the full force of the need is felt, firstly, to recruit those employees with specialist knowledge and, secondly and most importantly, to retain them and to capitalize on their potential. As the visionary noted [5], it is more difficult to lose human capital, when the firm develops its own specific abilities that are difficult to transfer. The negative consequences of that fact are accentuated, when key employees, leaders or those with a high performance are affected, because it is very difficult to replace workers with a high contribution without affecting global performance [6] [7].

In an effort to avoid these situations, the different configurations that motivate employees to remain in a firm are listed for review. Beforehand, it should be mentioned that the term configuration is linked to the configurational approach of Strategic Human Resource Management [8], according to which there is a coherent set of HR practices with strategic objectives that contribute to the competitiveness and the survival of the firm. Thus, the adjustment between employee and organization is quite a good predictor of the intention to remain committed to the firm or to leave it [9], such that high levels of adjustment are more likely to lead to the subsequent retention of employees. Both, competitive salaries as well as systems of financial rewards, are the most significant factors in the decision to leave the organization or to seek new opportunities [10], [11].

High salaries can act as a signal to retain qualified employees, protecting the firm from losing its investments in training, and avoiding extra expenditure on vacancies caused by departures [12]. Furthermore, high levels of retention are achieved through the offer of a salary, linked to other investments, such as the creation of structures to help new workers to acclimatize to their jobs and their environment, efforts to reduce unequal treatment between workers, and a commitment to small but incremental opportunities for development and progress [13]. [14] showed that the announcement of plans for share options by organizations slowed voluntary departures.

Moreover, certain environments are more favorable for employee retention than others [15]. Thus, young workers demand flexible and informal working environments [16]. Employees may be less willing to leave an organization with a system of social practices that reinforces company identity. The creation of a strong social atmosphere helps to encourage companionship between employees, leading to low rates of abandonment, as they feel reluctant to leave their friends [17], because a sense of belonging is reinforced [18] and, equally, because a high social implication in the organization is positively related to commitment [19] [20] [21] [22].

It appears clear that the more satisfied and the more motivated an individual feels, the more difficult it will be for that individual to leave the organization; the concept of commitment could be extended as far as the degree to which employees see the organization as a source of satisfaction [23]. Relations with companions are, generally, linked to on-the-job satisfaction, which is in turn related to a sense of belonging to the firm [24] [25]. A culture that emphasizes interpersonal relations will

attract professionals, more than another that is characterized by the value of its tasks [26]. The experiment conducted by [27], with a group of key employees, showed how they really consider themselves as relevant members when they belonged to a working environment with a high degree of interaction between people; when the work gave them autonomy, challenges, feedback, opportunities for development and the possibility to demonstrate their skills; and, also, if there was provision for continuous training and education.

Therefore, the firm can significantly increase its employee retention rate, deepening both satisfaction and motivation [28] [29]; at the same time as reducing the costs of employee departures [30] [31], as it has been demonstrated that both labor and extra-labor aspects influence the decision to remain in the firm [32]. In consequence, the configuration of employee retention, with what has been shown above, includes the components associated with employee adjustment to the organization, above-average remuneration in relation to competitors, harmonization with organizational culture, investments and support for recent arrivals, and social training events and actions.

Based on the ideas discussed above, the authors conducted a broad analysis of factors [33], ranging across advanced HR practices to explanations of firm performance. In doing so, our study [33] focused on intermediate indicators, such as employee characteristics, organizational capabilities and some other internal features. The factors in the study consisted of five settings for HR practices (acquisition, development, commitment, retention and flexibility), five employee features (human, social capital, organizational capital, motivation and turn over), four organizational capabilities (knowledge creation, knowledge application, organizational flexibility and information technologies), and some other internal features (strategic vision, HR emphasis, heterogeneity, and task-associated technology). Only those decisions relating to the acquisition of specialized personnel and employee-retention rates and the configuration of those two points (Strategic HR Management [34]) were analyzed.

Going one step further, the present research thoroughly analyzed those features related to Knowledge Retention policies. To do so, neural projection models [1] [35] [36], described in section 2, were trained and applied to generate intuitive visualizations of the data by reducing dimensionality. The instances are depicted in different colors and symbols to help extract conclusions, incorporating further meaningful information into those projections. Further details about the experiments are provided in section 3. Section 4 presents the obtained results while conclusions and future work are outlined in section 5.

2 Neural Projection Model

Projection models [37] operate on the spatial coordinates of high-dimensional data, in order to project them onto lower dimensional spaces. The main goal is to identify the patterns that exist across dimensional boundaries by identifying “interesting” directions, in terms of any specific index or projection. Such indexes or projections are, for example, based on the identification of directions that account for the largest

variance of a data set –i.e. Principal Component Analysis (PCA) [38], [39]- or the identification of higher-order statistics such as the skew or kurtosis index –i.e. Exploratory Projection Pursuit (EPP) [37]. Having identified the most interesting projections, the data is then projected onto a lower dimensional subspace, plotted onto two or three dimensions, which makes it possible to examine its structure with the naked eye.

The combination of projection techniques together with the use of scatter plot matrices is a very useful visualization tool to investigate the intrinsic structure of multidimensional data sets, allowing experts to study the relations between different components, factors or projections, depending on the technique that is applied.

The solution proposed in this research applies an unsupervised neural model called Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [2] [35]. It is based on Maximum Likelihood Hebbian Learning (MLHL) [2] [35], and introduces the application of lateral connections [2] [35] derived from the Rectified Gaussian Distribution [40]. This connectionist model has been chosen because it reduces the data dimensionality while preserving the topology in the original data set. Considering an N-dimensional input vector (x), and an M-dimensional output vector (y), with W_{ij} being the weight (linking input j to output i), then CMLHL can be expressed as:

1. Feed-forward step:

$$y_i = \sum_{j=1}^N W_{ij} x_j, \forall i . \tag{1}$$

2. Lateral activation passing:

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ . \tag{2}$$

3. Feedback step:

$$e_j = x_j - \sum_{i=1}^M W_{ij} y_i, \forall j . \tag{3}$$

4. Weight change:

$$\Delta W_{ij} = \eta \cdot y_i \cdot \text{sign}(e_j) |e_j|^{p-1} . \tag{4}$$

Where: η is the learning rate, τ is the “strength” of the lateral connections, b the bias parameter, p a parameter related to the energy function and A a symmetric matrix used to modify the response to the data [2] [35]. The effect of this matrix is based on the relation between the distances separating the output neurons. This neural projection model has been previously applied to the KM field [1].

3 Experimental Study

In its empirical validation of acquisition and retention factors, this study looked at 126 high-tech organizations in Spain. 267 R&D employees from these firms were surveyed, in order to analyze their HR strategies and subsequently improve the status of the analyzed firms.

The typical profile of these hi-tech firms would be an organization with 266 employees, manufacturing products and services (111 from the 126 organizations under study). A total of 47% of firms claimed that they innovated in both products and services, running 124 annual R&D programs. A total of 44% of the firms were members of a corporate group, 16% of which were international.

As only the HR retention factors are considered in the present study, features relating to these factors are analyzed in the surveyed data and described in Table 1.

Table 1. Analyzed components for retention factor

Retention factor
C1: Candidates are selected according to their fitting with the firm.
C2: Employees match the organizational culture.
C3: New employees are supported.
C4: Social and outdoor activities are sponsored by the firm for employees to know each other.
C5: Higher salaries than competitors are offered to retain employees.

Five components were used to define the retention factor. All these features have discrete values that range from 1 (strongly low) to 5 (strongly high). As a result, five features from each of the (126) firms were collected for the dataset.

Two analyses were done for the purpose of an interesting comparison. In the first one, each component or retention criteria was weighted with an identical value (0.20), which established a typology or categorization of firms in the category as: Excellent, High, Normal, Low and Poor. An “excellent” firm in the retention configuration means that all components obtained a score of between 4.5 and 5. The firms that fell into the “high” category scored between 3.5 and 4.4. in all criteria; while the “normal” consideration was for companies with values of between 2.8 and 3.4; the “low” category for scores between 2 to 2.7 and the “poor” category, whenever the score was below 2.

The second analysis consisted in assigning a greater weight to one of the components (0.4), leaving the rest equal (0.15), and so on, respectively, for each of the criteria. The aim is, in this way, to evaluate the overall consequences and effects that each component has on the knowledge retention factor, visualizing the variations that are shown. The results of this comparative study help us determine the specific measures that the firm should take, in areas such as employee selection, actions to encourage employees to identify with the organizational culture, protocols and measures for the integration of new employees and components to determine attractive salaries. All of these to achieve efficiency in the configuration of HR retention in the firm.

4 Results and Analysis

This section comprises an analysis of the best projections obtained in the above-described experimental study by applying the CMLHL model to the data on the knowledge retention factor. Fig. 1 shows the groups with their labels (1.1, 1.2 ...), projected by means of CMLHL and according to the retention factor.

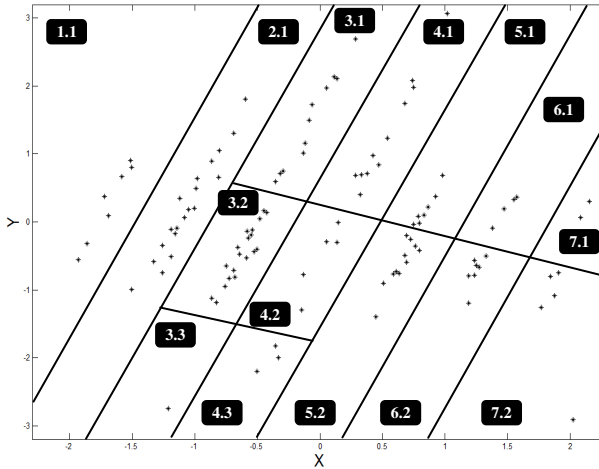


Fig. 1. CMLHL projection of the Retention Factor dataset

4.1 Analysis 1

Firstly, the combination of CMLHL and glyph depiction was applied to the equally-weighted features of knowledge retention. With regard to the first study, the application of the tool to the data and its subsequent interpretation immediately identified the different situations that can arise in the configuration of retention. In consequence, it provides information to take well-judged decisions for employee retention in firms with similar characteristics to those in this study. It is a preliminary step of transcendental importance to achieve the business objectives of competitiveness and survival.

Analyzing the five components all together (Fig. 2), the firms that are found to the right of the diagram present excellent knowledge retention, a position that starts to worsen as we move towards the left, such that those situated in the center represent high retention, which becomes normal as we move in that same direction, until we arrive at the low and poor categories in the positions furthest to the left. Simultaneously, the tool also shows the excellent firms situated in the lower part of the diagram, the high firms in the intermediate zone, and the remainder in the upper zone.

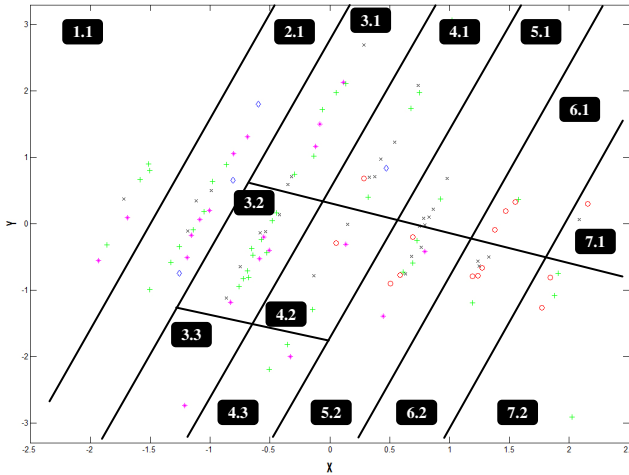


Fig. 2. CMLHL projection of the Retention Factor dataset for equally-weighted features

The grid generated by the application of the tool is very useful to classify firms and to place them with respect to the configuration of knowledge retention and, in consequence, to adopt the decisions that contribute to maintaining and to correcting these locations. In other words, an acceptable interpretation of the relevant data will highlight the existing configuration and will contribute sufficient knowledge, if necessary, in order to change position, determining not only the components that obtain little value, but also stressing the measures to be able to change undesirable locations, or in other words, firms in weaker categories. These measures, in R&D companies are connected with the selection of employees in accordance with the global adjustment of the firm, its clear identification with organizational culture, integration, the development of activities for recently contracted employees and a higher salary offer than its competitors.

4.2 Analysis 2

Fig. 3 shows the CMLHL projection of the analysis of firms, according to the retention factor, varying the component weights. The overvalued factor is marked with a plus character ('+'). That is, "+C1" means that the component 1 (See Table 1) was given a higher weight than others.

In addition to the interpretation of the positions that the firms occupy in the projection (Fig. 3), the study was complemented by a second analysis, with the objective of comparing the variation or effect of each component on the overall retention factor. To do so, each component of the retention configuration, starting with the first one and finishing with the fifth one, were iteratively weighted with a higher value (0.50) with regard to the others (0.15 for each of the four remaining ones) and its effect on the initial typology was studied (where all the criteria were weighted with an identical value). In this way, five different experiments were conducted, varying the weight from the first to the fifth component.

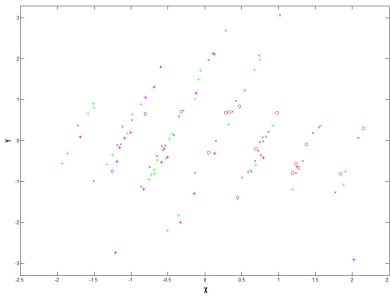


Fig. 3.1. CMLHL projections of the retention factor dataset: overweighted component 1 (+C1)

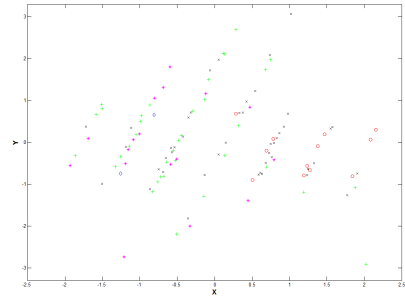


Fig. 3.2. CMLHL projections of the retention factor dataset, overweighting component 2 (+C2)

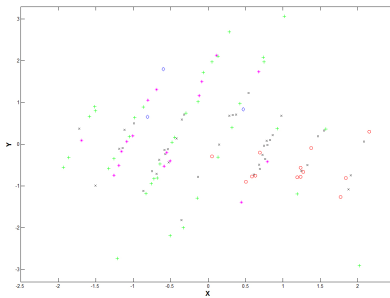


Fig. 3.3. CMLHL projections of the retention factor dataset, overweighting component 3 (+C3)

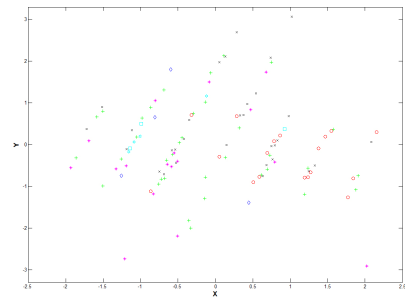


Fig. 3.4. CMLHL projections of the retention factor dataset, overweighting component 4. (+C4)

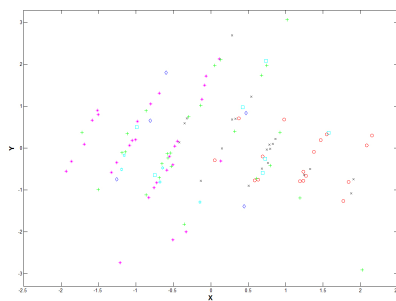


Fig. 3.5. CMLHL projections of the retention factor dataset, overweighting component 5 (+C5)

The results were very revealing. Starting with Fig. 2, fourteen clearly identified groups or sections were observed, each of which represents, according to this study, two revealing aspects of the firms in the sample, their category or the level that they reach in the retention configuration, as well as the high values that correspond with excellent or very good human resource retention, while the very low values represent the contrary situation, and their localization.

When the highest weighting is attached to component C1 (selection of individuals according to the global adjustment with the firm), the results present the following changes (Fig. 3.2 +C1): in group 1.1, one firm moves from a low to a normal category; in 2.1, one firm changes from poor to low and two others change from normal to high categories; in group 3.1, one firm changes from low to normal, two from normal to high and one from high to excellent; in 3.2, two firms move from normal to high; in section 4.1, one firm changes from normal to high and one more from high to excellent; in 4.2, one moves from normal to high; in group 5.1, one passes from high to excellent; in group 5.2, another changes from normal to high; in group 6.2 as well as in group 7.2, a single firm changes from high to excellent. There are a total of sixteen changes, in all of which the firms improve the retention configuration with regard to their initial situation.

In the case of the greater weighting given to C2 (employees identify with the organizational culture), the variations are (Fig. 3.2 +C2) as follows: in group 2.1, one firm moves from poor to low and another from normal to high; in 3.1, three firms vary from low to normal, while another moves from normal to high; in 3.2, one firm changes from low to normal, while another three move from normal to high; in group 4.1, one firm changes from poor to low and another from normal to high; in 4.2, one moves from low to normal, another from normal to high, and yet another from high to excellent; in group 5.1, one company switches the category of normal for high, and another switches from high to excellent; in group 5.2, two firms change from normal to high; in group 6.1, one company moves from normal to high; in 6.2, another moves from high to excellent, in 7.1, one firm changes from high to excellent and in group 7.2, one firm passes from normal to high. The changes affect twenty-three firms, all of which improve the retention configuration, as in component C1.

C3 (the firm provides support for the incorporation of recently contracted employees) is given a greater weighting in retention, but with the following changes (Fig. 3.2 +C3): in group 1.1, one single firm moves from low to normal; in 2.1, one firm changes from poor to low, and two others from normal to high; in group 3.2, one firm changes from low to normal and four more from the normal to the high category; group 4.2 presents a change from low to normal; in 4.3, one changes from poor to normal and another from normal to high; in group 5.2, three companies change from normal to high and another from high to excellent; finally, in group 6.2, one firm changes from high to excellent. Here too, the changes are improvements for all sixteen firms.

With regard to attributing greater weight to component C4 (promotion of social events and activities so that workers relate and get to know each other), the results observed are as follows (Fig. 3.2 +C4):

In group 1.1, four firms change from normal to low and one other from high to normal; in 2.1, another four firms move from normal to low and two more from high to normal; in 3.1, one company changes from normal to high, in 3.2, six firms change from normal to low and three more from high to low; in group 4.1, two companies switch from the high to the normal category and another one from excellent to high; in 4.2, one moves from normal to low, and the same happens in 4.3; in 5.2, one firm changes from low to poor and another from excellent to high. In view of these data, criterion C4 is clearly the only one that provokes more unfavorable or retrograde situations, affecting 38 companies, at the same time as it is shown to be the component that provokes the most changes in the firms and is the one that differentiates or discriminates between the data under analysis more than any other. This is because the firms under study clearly present inferior values under this criterion to those of the other components, except in groups 6.1, 6.2, 7.1 and 7.2 that reach the highest possible value.

Finally, when C5 (offers higher salaries than competitors) is given a greater weighting, the following transformations appear (Fig. 3.2 +C5): in 1.1, one firm moves from normal to high; in 2.1., another firm varies from low to normal; in 3.1., one firm moves from low to normal, another two from normal to high and one from high to excellent; in group 3.2, two firms move from normal to high and one from high to excellent, in 4.1, one firm changes from poor to low and another from normal to high; in 4.3, one moves from low to normal; in group 5.1, two entities are transformed from high to excellent. In this component, 15 changes of firms occur, which improve the level of their human resources retention factor.

These results demonstrate that the impact of each component on the retention configuration may be understood and, in consequence, strategic decisions may be taken according to the objectives that are pursued. Thus, the question is how to influence those behaviours and actions, of which the firm may be unaware or may be ignoring, once they have been identified in the analysis. In concrete, there is a wide range of possibilities that runs from designing and applying selection policies that take into account the needs for knowledge within the firm, until the actions are proposed that allow the employees to become aware of, to understand, and to make the value of the organizational culture their own.

Likewise, the firm should be aware that the retention configuration calls for the provision of help so that the incorporation of new employees is a success, as well as for meetings and events to promote relations between veteran staff and recent arrivals. In this sense, integration policies should be based on the particularities of the people, knowledge exchange and sharing, the establishment of teams of people with different levels experience and service and the context in which the work will develop. Finally, as made clear, the policies of remuneration and motivation are of vital importance to achieve an excellent level of human resource retention.

A comparison of CMLHL projections with those obtained from some other unsupervised techniques: Principal Component Analysis (PCA) [39], MLHL and Self-Organizing Map (SOM) [41] were also applied to the HR dataset previously analyzed. Only those projections obtained by CMHL have been included in present study as they are more sparse [2] [35] and the data ordering can be identified in a more clear way.

5 Conclusions and Future Work

The objective that prompted us to conduct this study was twofold: on the one hand, to prove the validity of the technique, for a set of qualitative, abstract and disordered data, representative of the level reached in the retention configuration of R&D personnel, in a significant sample of Spanish high-tech firms (126), and, on the other hand, to contribute a necessary element of rigor and robustness to research in the field of KM. Accordingly, it was possible to scale up from theoretical formulations to the application tools, the purpose of which is first of all to diagnose the reality and, then, to shed light on or to guide certain actions that improve their application or implementation in the firm.

This objective may be said to have been achieved because the application of the tool has served not only to group together and to order the data, the interpretation of which is highly significant, but also to arrive at interesting results, which serve to take decisions that can assure the effectiveness of the knowledge retention configuration. This HR practice has a high-level impact on competitiveness and the survival of the firm. From among these results, the following may be highlighted:

1. Clear identification of the firm's position with regard to its employee retention factor, which at the same time emerges from the configuration components attributed to it. This allows us to represent and, in consequence, to interpret the importance or commitment that the firms in this study attribute to this point, representative of a working base with the necessary knowledge and capabilities, whether economic, sustainable and stable, and in which the best professionals from each area are concentrated and maintained. Simultaneously, together with the preceding point on the position of the firm, the representation of the components or the variables of this factor provide information on the category or level in each configuration component and, therefore, help determine which achieve the best values and in which intervention is necessary. More specifically, the results allow us to establish, for example, if the firm has a plan where the needs of specific abilities are specified for the available posts and if higher salaries are offered to contract the best worker, where this salary is an element of retention. Likewise, it establishes if the selection is done on the basis of the global adjustment of the firm, if the employees identify with the organizational culture and whether there is a plan for the integration of new employees. All these elements work towards the success of human resource retention in the firm.
2. The convergence of various variables, each one with identical weightings, in clearly identified areas or spaces, which represent business positions or situations with regard to different levels of the retention factor; descending from excellent, high, and normal, to low and then poor. And their comparison, when the weighting is different for each of the five variables under analysis, maintaining the rest respectively equal. The displacement of the data set to other slightly different zones, as well as its variation in the primary score that was given, establish the characteristics of the firm, and those that it can manage to achieve, according to the actions that it undertakes, to situate itself, according to each case, in the best positions, changing from one to another, according to the employee retention strategy that is needed or relinquishing strategies that have proved inefficient.

In brief, the tool applied to the data used in this analysis has to a high degree of accuracy detected the position, the level and the components that configure working practice for the retention of R&D employees in a significant sample of Spanish firms, and the way in which it may evolve and progress towards a more effective introduction and execution, which will achieve higher levels of competitiveness and value. Acceptable behavior of the tool, in this and in earlier works, opens a wide field of possibilities for data treatment in fields where these are of a qualitative, abstract, and disordered nature, which will without doubt contribute to improving their diagnosis and may, more importantly, serve to advance applied research, providing guidance to business managers and executives.

Future work will focus on extending this study to some other factors that are important in the HR and KM fields (described in Section 2) and on the application of other unsupervised visualization models.

Acknowledgments. This research was partially supported through grants awarded by the Spanish Ministry of Economy and Competitiveness (ref: TIN2010-21272-C02-01) funded by the European Regional Development Fund, and by the Junta de Castilla y León (ref: SA405A12-2).

References

1. Herrero, Á., Corchado, E., Sáiz, L., Abraham, A.: DIPKIP: A Connectionist Knowledge Management System to Identify Knowledge Deficits in Practical Cases. *Computational Intelligence* 26(1), 26–56 (2010)
2. Durst, S., Edvardsson, I.R.: Knowledge Management in SMEs: a Literature Review. *Journal of Knowledge Management* 16(6), 879–903 (2012)
3. Pandey, S.C., Dutta, A.: Role of Knowledge Infrastructure Capabilities in Knowledge Management. *Journal of Knowledge Management* 17(3), 435–453 (2013)
4. Levy, M.: Knowledge Retention: Minimizing Organizational Business Loss. *Journal of Knowledge Management* 15(4), 582–600 (2011)
5. Becker, G.S.: *Human Capital*. Columbia University Press (1964)
6. Jiang, K., Lepak, D.P., Han, K., Hong, Y., Kim, A., Winkler, A.-L.: Clarifying the Construct of Human Resource Systems: Relating Human Resource Management to Employee Performance. *Human Resource Management Review* 22(2), 73–85 (2012)
7. Mowday, R.T.: Reflections on the Study and Relevance of Organizational Commitment. *Human Resource Management Review* 8(4), 387–401 (1999)
8. Kalchschmidt, M.: Best Practices in Demand Forecasting: Tests of Universalistic, Contingency and Configurational Theories. *International Journal of Production Economics* 140(2), 782–793 (2012)
9. Boon, C., Den Hartog, D.N., Boselie, P., Paauwe, J.: The Relationship between Perceptions of HR Practices and Employee Outcomes: Examining the Role of Person–organisation and Person–job Fit. *The International Journal of Human Resource Management* 22(01), 138–162 (2011)
10. Tremblay, M., Vandenberghe, C., Doucet, O.: Relationships Between Leader-Contingent and Non-contingent Reward and Punishment Behaviors and Subordinates' Perceptions of Justice and Satisfaction, and Evaluation of the Moderating Influence of Trust Propensity, Pay Level, and Role Ambiguity. *Journal of Business and Psychology* 28(2), 233–249 (2013)

11. Tremblay, M., Cloutier, J., Simard, G., Chênevert, D., Vandenberghe, C.: The Role of HRM Practices, Procedural Justice, Organizational Support and Trust in Organizational Commitment and In-role and Extra-role Performance. *The International Journal of Human Resource Management* 21(3), 405–433 (2010)
12. Aguinis, H., Joo, H., Gottfredson, R.K.: What Monetary Rewards Can and Cannot Do: How to Show Employees the Money. *Business Horizons* 56(2), 241–249 (2013)
13. Lam, W., Chen, Z., Takeuchi, N.: Perceived Human Resource Management Practices and Intention to Leave of Employees: the Mediating Role of Organizational Citizenship Behaviour in a Sino-Japanese Joint Venture. *The International Journal of Human Resource Management* 20(11), 2250–2270 (2009)
14. Gifford, R.H.: An Analysis of the Market Response to Announcements of Broad-based Stock Option Plans and an Analysis of the Effects of Broad-based Plans on Firm Performance, Employee Behavior and Employee Retention. University Microfilms International (2002)
15. Širca, N.T., Babnik, K., Breznik, K.: Towards Organisational Performance: Understanding Human Resource Management Climate. *Industrial Management & Data Systems* 113(3), 367–384 (2013)
16. Haesli, A., Boxall, P.: When Knowledge Management Meets HR Strategy: an Exploration of Personalization-retention and Codification-recruitment Configurations. *The International Journal of Human Resource Management* 16(11), 1955–1975 (2005)
17. Ellinger, A.E., Musgrove, C.C.F., Ellinger, A.D., Bachrach, D.G., Elmadağ Baş, A.B., Wang, Y.-L.: Influences of Organizational Investments in Social Capital on Service Employee Commitment and Performance. *Journal of Business Research* (2012)
18. Mamman, A., Kamoche, K., Bakuwa, R.: Diversity, Organizational Commitment and Organizational Citizenship Behavior: An Organizing Framework. *Human Resource Management Review* 22(4), 285–302 (2012)
19. Zatzick, C.D., Iverson, R.D.: Putting Employee Involvement in Context: a Cross-level Model Examining Job Satisfaction and Absenteeism in High-involvement Work Systems. *The International Journal of Human Resource Management* 22(17), 3462–3476 (2011)
20. Wood, S., Van Veldhoven, M., Croon, M., de Menezes, L.M.: Enriched Job Design, High Involvement Management and Organizational Performance: The Mediating Roles of Job Satisfaction and Well-being. *Human Relations* 65(4), 419–445 (2012)
21. Herold, D.M., Fedor, D.B., Caldwell, S., Liu, Y.: The Effects of Transformational and Change Leadership on Employees' Commitment to a Change: A Multilevel Study. *Journal of Applied Psychology* 93(2), 346–356 (2008)
22. de Menezes, L.M., Wood, S., Gelade, G.: The Integration of Human Resource and Operation Management Practices and its Link with Performance: a Longitudinal Latent Class Study. *Journal of Operations Management* 28(6), 455–471 (2010)
23. Mowday, R.T., Steers, R.M., Porter, L.W.: The Measurement of Organizational Commitment. *Journal of Vocational Behavior* 14(2), 224–247 (1979)
24. Clark, K., Lengnick-Hall, M.L.: MNC Practice Transfer: Institutional Theory, Strategic Opportunities and Subsidiary HR Configuration. *The International Journal of Human Resource Management* 23(18), 3813–3837 (2012)
25. Kehoe, R.R., Wright, P.M.: The Impact of High-Performance Human Resource Practices on Employees' Attitudes and Behaviors. *Journal of Management* 39(2), 366–391 (2013)
26. Meyer, J.P., Hecht, T.D., Gill, H., Toplonysky, L.: Person–organization (Culture) Fit and Employee Commitment under Conditions of Organizational Change: a Longitudinal Study. *Journal of Vocational Behavior* 76(3), 458–473 (2010)

27. Brown, M.M.: An Exploratory Study of Job Satisfaction and Work Motivation of a Select Group of Information Technology Consultants in the Delaware Valley. Wilmington College (2002)
28. Reiche, B.S.: The Configuration of Employee Retention Practices in Multinational Corporations' Foreign Subsidiaries. *International Business Review* 17(6), 676–687 (2008)
29. Shaw, D., Edwards, J.S.: Building User Commitment to Implementing a Knowledge Management Strategy. *Information & Management* 42(7), 977–988 (2005)
30. Parden, R.J.: The Manager's Role and the High Mobility of Technical Specialists in the Santa Clara Valley. *IEEE Transactions on Engineering Management* (1), 2–8 (1981)
31. McClean, E., Collins, C.J.: High-commitment HR Practices, Employee Effort, and Firm Performance: Investigating the Effects of HR Practices across Employee Groups within Professional Services Firms. *Human Resource Management* 50(3), 341–363 (2011)
32. Nishii, L.H., Lepak, D.P., Schneider, B.: Employee Attributions of the “why” of HR Practices: Their Effects on Employee Attitudes and Behaviors, and Customer Satisfaction. *Personnel Psychology* 61(3), 503–545 (2008)
33. Sáiz, L., Pérez, A.: Formación de las Capacidades de Creación de Conocimiento y Flexibilidad Organizativa en Empresas de Alta Tecnología. In: 4th International Conference on Industrial Engineering and Industrial Management, pp. 800–809 (2010)
34. Dolan, S.L., Mach, M., Olivera, V.S.: HR Contribution to a Firm's Success Examined from a Configurational Perspective: An Exploratory Study Based on the Spanish CRANET Data. *Management Review* 16(2), 272 (2005)
35. Herrero, Á., Corchado, E., Jiménez, A.: Unsupervised Neural Models for Country and Political Risk Analysis. *Expert Systems with Applications In Press* (2011)
36. Herrero, Á., Corchado, E., Gastaldo, P., Zunino, R.: Neural Projection Techniques for the Visual Inspection of Network Traffic. *Neurocomputing* 72(16-18), 3649–3658 (2009)
37. Friedman, J.H., Tukey, J.W.: A Projection Pursuit Algorithm for Exploratory Data-Analysis. *IEEE Transactions on Computers* 23(9), 881–890 (1974)
38. Hotelling, H.: Analysis of a Complex of Statistical Variables into Principal Components. *Journal of Education Psychology* 24, 417–444 (1933)
39. Pearson, K.: On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine* 2(6), 559–572 (1901)
40. Seung, H.S., Soccì, N.D., Lee, D.: The Rectified Gaussian Distribution. *Advances in Neural Information Processing Systems* 10, 350–356 (1998)
41. Kohonen, T.: The Self-Organizing Map. *Proceedings of the IEEE* 78(9), 1464–1480 (1990)

Fall Detection Using Kinect Sensor and Fall Energy Image

Bogdan Kwolek¹ and Michal Kepski²

¹ AGH University of Science and Technology, 30 Mickiewicza Av.,
30-059 Krakow, Poland

`bkw@agh.edu.pl`

² University of Rzeszow, 16c Rejtana Av., 35-959 Rzeszów, Poland
`mkepski@univ.rzeszow.pl`

Abstract. One of the main reasons for low acceptance by seniors the available technology for automatic fall detection is that the existing devices generate too much false alarms. Additionally, the camera-based devices do not preserve the privacy adequately. In our approach an accelerometer is utilized to indicate a potential fall. A fall hypothesis is then verified in the second stage in which we employ a depth image, which was shot at the moment of the potential fall. A detector that was trained in advance on features extracted both from depth images and points cloud is responsible for verification whether a person is lying on the floor. After all, to reliably distinguish the fall from fall-like activities we perform final verification, in which we employ the proposed fall energy image. The fall energy image expresses the distribution of the person's motion in the set of images preceding the fall.

Keywords: Depth image and point cloud processing, fall detection.

1 Introduction

Falls are a major health risk and a significant obstacle to independent living of the seniors [13]. In response to the demand for fall detection technology, plenty of research has been done in the recent years to develop unobtrusive fall detection systems for enhancing the functional ability of the elderly and patients [16]. However, despite many efforts made to obtain reliable and unobtrusive person fall detection, current technology does not meet the seniors' needs. One of the main reasons for non-acceptance of the currently available technology by elderly is that the existing devices generate too much false alarms, which in turn lead to considerable frustration of the seniors. Additionally, the existing devices do not preserve the privacy and unobtrusiveness adequately.

In recent years, a lot of research has been done on detecting falls using a wide range of sensor types. Mubashir et al. [16] done a survey of methods used in the existing systems. Single CCD camera [18], multiple cameras [6], specialized omni-directional ones [15] and stereo-pair cameras [9] are widely used in the vision systems for fall detection. Most of the currently available techniques for fall

detection are based on wearable sensors. Accelerometers or both accelerometers and gyroscopes are the most frequently used sensors in devices responsible for fall monitoring [17]. However, on the basis of inertial sensors it is not easy to separate real falls from fall-like activities [2] and in consequence the devices that are built on only such sensors typically trigger significant amount of false alarms. The reason is that the characteristic motion patterns of fall also exist in many actions. For instance, the crouch also demonstrates a rapid downward motion.

Recently, Kinect sensor was used in prototype systems for fall detection [10, 11, 14]. It is the world's first low-cost device that combines an RGB camera and a depth sensor. Unlike 2D cameras, it allows 3D tracking of the body movements. Thus, if only depth images are used it preserves the person's privacy. Because depth images are extracted with the support of an active light source, they are largely independent of external light conditions. Thanks to the use of the infrared light the Kinect is capable of extracting the depth images in dark rooms.

In this work we demonstrate an approach to reduce the number of false positives alarms in fall detection through the use of an accelerometer and the depth images. The accelerometer is utilized to indicate a potential fall. A fall hypothesis is then verified in the second stage in which we employ a depth image, which was shot at the time of the potential fall of the person. A detector that was trained in advance on features extracted both from depth images and points cloud is responsible for verification whether a person is lying on the floor. After all, to reliably distinguish the fall from fall-like activities we perform final verification, in which we employ the proposed fall energy image. The fall energy image expresses the distribution of the person's motion in a collection of the images, acquired in a certain period of time before the potential fall alert.

The contribution of this work is twofold: firstly we propose fall energy images (FEI) as an effective spatiotemporal representation of the human fall. Secondly, we show how to extract such energy fall images on the basis of the depth images and then how to utilize them to achieve reliable fall detection. Shape modeling using spatiotemporal features provides crucial information about human activities. In [7], a method for fall detection that is based on a combination of the eigenspace and integrated time motion images (ITMI) was developed. ITMI contain motion information and time stamps of motion occurrence. Multilayer perceptron neural network was utilized for classification of motions and detection of the fall event. In [19], a mobile human airbag system was designed for fall protection for the elderly. A Micro Inertial Measurement Unit consisting of three dimensional accelerometers, gyroscopes, a Bluetooth module and a Micro Controller Unit (MCU) is utilized to record human motion information. Through analysis of images acquired by a high-speed camera, a lateral fall can be determined on the basis of a gyro threshold. The classification of falls is performed by a support vector machine (SVM) classifier. The majority of vision based systems for fall detection do not take into account the motion information. In this work we demonstrate how to extract fall energy images using accelerometer and depth images as well as how to process them. The accelerometer helps us to extract the representative segment of the images as a representation of the fall event.

2 Person Detection in Depth Images

Depth is very useful cue to achieve reliable person detection because humans may not have consistent color and texture but have to occupy an integrated region in space. The Kinect combines structured light with two classic computer vision techniques, namely depth from focus and depth from stereo. It is equipped with infrared laser-based IR emitter, an infrared camera and a RGB camera. The IR camera and the IR projector compose a stereo pair with a baseline of approximately 75 mm. A known pattern of dots is projected from the IR laser emitter. These specs are captured by the IR camera and then compared to the known pattern. Since there is the distance between laser and sensor, the images correspond to different camera positions, and that in turn allows to use stereo triangulation to calculate each spec depth. The field of view of the system is 57° horizontally and 43° vertically, the minimum measurement range is about 0.6 m, whereas the maximum range is somewhere between 4-5 m. It captures the depth and color images simultaneously at a frame rate of about 30 fps. The default RGB video stream has size 640×480 and 8-bit for each channel, whereas the depth stream is 640×480 resolution and with 11-bit depth.

The software called NITE from PrimeSense offers skeleton tracking on the basis of depth images. However, this software is targeted for supporting the human-computer interaction, and not for detecting the person fall. Thus, in many circumstances it can have difficulties in extracting and tracking the person's skeleton. Therefore, we employ a person detection method [11], which reliably extracts the subject including situations when he/she is lying on the floor.

The person was delineated on the basis of a scene reference image, which was extracted in advance and then updated on-line. In the depth reference image each pixel assumes the median value of several pixels values from the past images. In the setup phase we collect a number of the depth images, and for each pixel we assemble a list of the pixel values from the former images, which is then sorted in order to extract the median. Given the sorted lists of pixels the depth reference image can be updated quickly by removing the oldest pixels and updating the sorted lists with the pixels from the current depth image and then extracting the median value. We found that for typical human motions, good results can be obtained using 13 depth images [11]. For Kinect acquiring the images at 25 Hz we take every fifteenth image.

In the detection mode the foreground objects are extracted through differencing the current depth image from such a depth reference map. Afterwards, the person is delineated through extracting the largest connected component in the thresholded difference between the current map and the reference map.

3 V-Disparity Based Ground Plane Extraction

Given a depth map provided by the Kinect sensor, the disparity d can be determined in the following manner:

$$d = \frac{b \cdot f}{z} \quad (1)$$

where z is the depth (in meters), b is the horizontal baseline between the cameras (in meters), f is the (common) focal length of the cameras (in pixels). The IR camera and the IR projector form a stereo pair with a baseline of approximately $b = 7.5$ cm, whereas the focal length f is equal to 580 pixels.

Let H be a function of the disparities d such that $H(d) = I_d$. The I_d is the v -disparity image and H accumulates the pixels with the same disparity from a given line of the disparity image. Thus, in the v -disparity image each point in the line i represents the number of points with the same disparity occurring in the i -th line of the disparity image. In [12] the v -disparity maps between two stereo images were used to achieve reliable obstacle detection. In our work the v -disparity maps are extracted using depth images determined by Kinect.

The line corresponding to the floor pixels in the v -disparity map was extracted using the Hough transform operating on v -disparity values and a predefined range of parameters. The accumulator was incremented by v -disparity values [11]. Assuming that the Kinect is placed at height about 1 m from the floor, the line representing the floor should begin in the disparities ranging from 21 to 25 depending on the tilt angle of the sensor. Given the extracted line in such a way, the pixels belonging to the floor areas were determined [11]. Due to the measurement inaccuracies, pixels falling into some disparity extent d_t were also considered as belonging to the ground. Assuming that d_y is a disparity in the line y , which represents the pixels belonging to the ground plane, we take into account the disparities from the range $d \in (d_y - d_t, d_y + d_t)$ as a representation of the ground plane.

After the transformation of the pixels representing the floor to the 3D points cloud, the plane described by the equation $ax+by+cx+d$ has been recovered [11]. The parameters a, b, c and d were estimated using the RANSAC algorithm. The distance to the ground plane from the 3D centroid of points cloud corresponding to the segmented person was determined on the basis of the following equation:

$$D = \frac{|aX_c + bY_c + cZ_c + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (2)$$

where X_c, Y_c, Z_c stand for the coordinates of the person's centroid.

4 Lying Pose Recognition

The recognition of lying pose was achieved using a classifier trained on features representing the extracted person both in depth images and in point clouds. A data-set consisting of images with normal activities like walking, sitting down, crouching down and lying has been composed in order to train a classifier responsible for testing whether a person is lying on the floor and to evaluate its performance. Thirty five volunteers with age under 28 years attended in preparation of the data-set. The image sequences were recorded using two Kinect devices. The first Kinect was placed at a height of about one meter to the floor, whereas the second one was placed at a ceiling corner of the room. Figure 1 shows example depth images seen from such two different views.

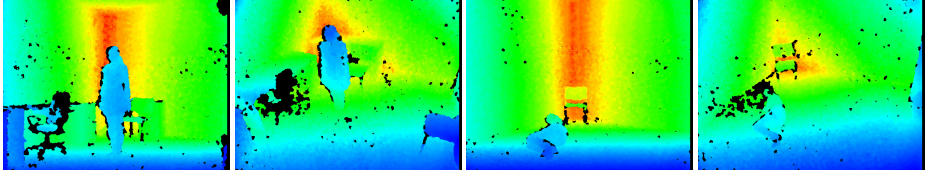


Fig. 1. Person in depth images seen from two different views

In total 312 images representing typical human actions were selected and then utilized to extract the following features:

- h/w - a ratio of width to height of the person’s bounding box, calculated in the points cloud
- h/h_{max} - a ratio expressing the height of the person’s surrounding box in the current frame to the height of the person
- $dist$ - the distance of the person centroid to the floor, expressed in millimeters
- $max(\sigma_x, \sigma_z)$ - standard deviation from the centroid for the abscissa and the depth, respectively.

Figure 2 depicts a scatterplot matrix for the employed attributes, in which a collection of scatterplots is organized in a two-dimensional matrix simultaneously to provide correlation information among the attributes. In a single scatterplot two attributes are projected along the x-y axes of the Cartesian coordinates. As we

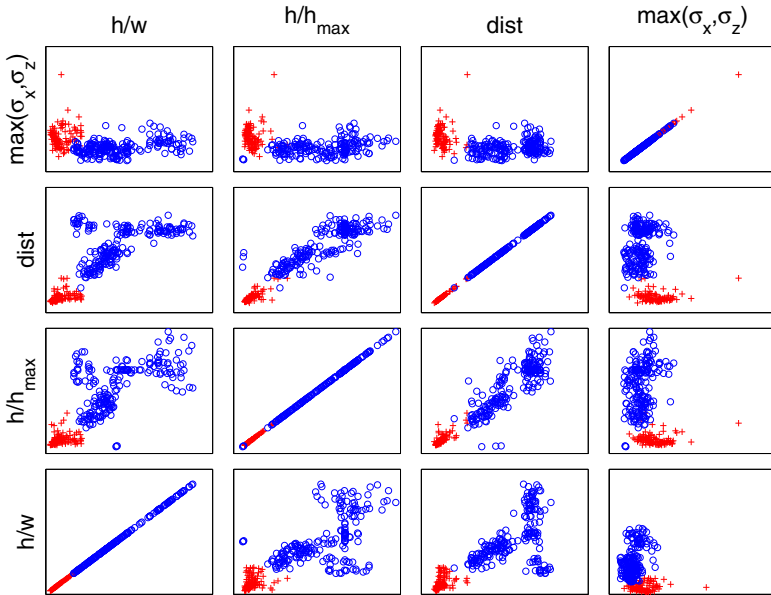


Fig. 2. Multivariate classification scatter plot for features used in lying pose recognition

can observe, the overlaps in the attribute space are not too significant. We considered also another attributes, for instance, a filling ratio of the rectangles making up the person's bounding box. The worth of the considered features was evaluated on the basis of the information gain [4], which measures the dependence between the feature and the class label. In the assessment of the discrimination power of the considered features and selecting the most discriminative ones we utilized the `InfoGainAttributeEval` procedure from the Weka [5], which is a collection of machine learning algorithms.

5 Fall Energy Image

Several motion features have been proposed until now to represent people activities, such as Motion History Image (MHI) [1]. Usually, the MHI is generated on the basis of binary images, where the person silhouette sequence is condensed into gray scale images as a weighted combination of all motion images. The result of such a motion condensation is a scalar-valued image in which more recently moving pixels are brighter. One of the advantages of the MHI representation is that a range of action images may be encoded in a single motion-shape. Typically, in action recognition phase such a static shape pattern is compared with pre-stored action prototypes.

The Fall Energy Image is an average of all silhouette images of a single fall. Such a spatiotemporal energy map spans the time scale of person fall. The energy map is calculated using a number of binary silhouette images before the fall. The images are scaled according to the distance of the person to the camera. We assume that a fall occurs if the signal upper peak value from the accelerometer is greater than $3g$. Figure 3 illustrates example fall energy images with the corresponding plots of signal upper peak value (UPV) vs. time. As we can observe, both actions have quite similar characteristics in the acceleration domain, but totally different fall energy maps.

The weighted average (moment) of the fall energy expressed by pixel intensities was computed using moments as follows:

$$\begin{aligned} x_c &= \frac{\sum_x \sum_y xP(x, y)}{\sum_x \sum_y P(x, y)} \\ y_c &= \frac{\sum_x \sum_y yP(x, y)}{\sum_x \sum_y P(x, y)} \end{aligned} \quad (3)$$

where x, y are pixel coordinates. The major length and width (eigenvalues) of the fall energy has been calculated in the following manner [8]:

$$\begin{aligned} l &= 0.707\sqrt{(a+c) + \sqrt{b^2 + (a-c)^2}} \\ w &= 0.707\sqrt{(a+c) - \sqrt{b^2 + (a-c)^2}} \end{aligned} \quad (4)$$

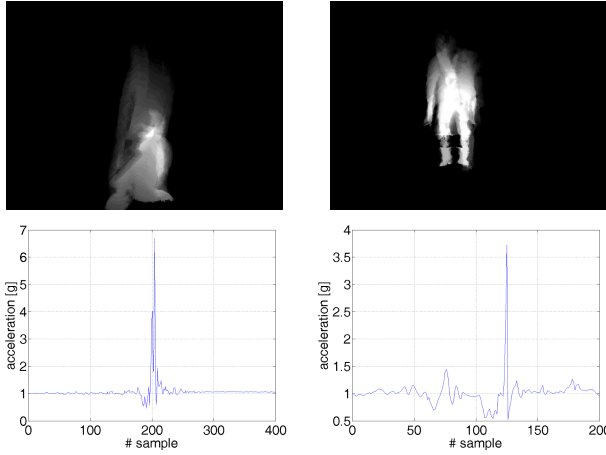


Fig. 3. Fall energy images for a forward fall (left) and sitting on a chair (right) with corresponding plots of signal upper peak value vs. time (bottom row)

where

$$a = \frac{M_{20}}{M_{00}} - x_c^2, \quad b = 2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right), \quad c = \frac{M_{02}}{M_{00}} - y_c^2, \quad M_{00} = \sum_x \sum_y P(x, y),$$

$$M_{11} = \sum_x \sum_y xyP(x, y), \quad M_{20} = \sum_x \sum_y x^2P(x, y), \quad M_{02} = \sum_x \sum_y y^2P(x, y).$$

We calculated also the average fall energy, i.e. the mean value of non-zero pixel values in the fall energy image $P(x, y)$ as well as the Euclidean distance d_E between the weighted average location of the fall energy (y_c, x_c) and the geometrical centroid of the thresholded energy map. Figure 4 depicts the scatter plot matrix for such energy features. The features were extracted on the basis of 30 image sequences in which half of them contained person falls. The remaining sequences contained person activities, which were very similar to fall. The activities were performed close to the floor and contained actions consisting in sitting on the floor, laying down on the floor, for instance to raise an object, etc. The features were extracted on the basis of 30 depth images just before the human fall, which in turn was signaled by a procedure processing data from the accelerometer. That means that the FEI image expresses the fall energy in about 1 sec. As we can observe, on the basis of such a set of features the person fall can be distinguished from the non-fall activities. We considered also energy features extracted on the basis of the bank of Log-Gabor filters. Their worth was evaluated on the basis of the information gain and then compared to the discrimination power of the above discussed features. The experimental results showed that their worth is not worse in comparison to Gabor filter based energy features and therefore we decided to use them in the evaluation of the whole system. It is worth to note that they can be extracted in considerably shorter time.

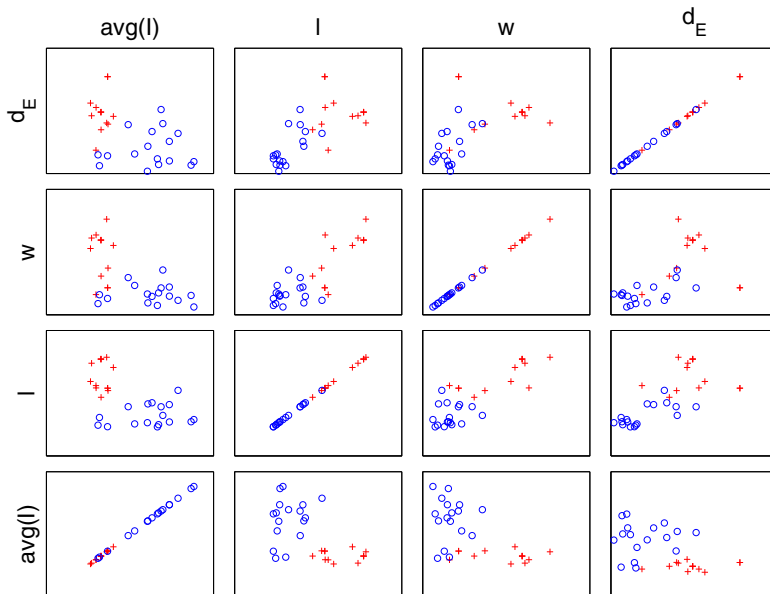


Fig. 4. Multivariate classification scatter plot for features extracted on fall energy images

6 Experimental Results

Thirty five young health volunteers with age under 28 years attended in preparation of the data-sets and in the evaluation of the fall detection accuracy. To show the resistance of the system to the placement of the camera the images were acquired by two Kinect devices. The motion data were acquired by a wearable smart device (Sony PlayStation Move) containing accelerometer and gyroscope sensors. Data from the device were transmitted wirelessly via Bluetooth and received by a laptop computer. In all, 312 images acquired by two Kinect devices were selected and then used to evaluate the k-NN classifier responsible for checking whether the person is lying on the floor. The number of images with a fall was equal to 110. We evaluated also KStar [3], SVM and multilayer perceptron (MLP) classifiers. The KStar and MLP classified all falls correctly, whereas the remaining algorithms incorrectly classified two instances. A k-NN based motion classifier was trained on 30 image sequences of which 15 contained fall events. Its accuracy was evaluated in 10-fold cross-validation and one fall was classified incorrectly. The SVM and KStar classified all falls correctly.

The complete system for fall detection was tested with simulated-falls performed by young volunteers under supervised conditions onto crash mats. The accelerometer was worn near the pelvis. Five volunteers attended in the tests and evaluations of our system. Intentional falls were performed in home towards a carpet with thickness of about 2 cm. Each individual performed ADLs like walking, sitting, crouching down, leaning down/picking up objects from the floor,

lying on a bed. As expected, using only the accelerometer the number of false alarms was considerable. Experimental results demonstrated that most of them can be ignored owing to the use of our recognition module of the lying pose. This operation is done at low computational cost as the verification of the fall is performed if the module processing the data from the accelerometer triggers the alarm. Moreover, on the basis of the accelerometer based alarm the system obtains information which image should be processed to decide if an event consisting in person lying on the floor takes place. All person activities that have been considered in the previous work [10] were classified correctly. During the evaluation of the system the volunteers found several fall-like actions, which were not considered in the previous work and for which the two-stage algorithm triggered false alarms. The experimental results obtained on the system with three modules, i.e. accelerometer, lying pose recognition and fall energy analysis demonstrated that the fall energy features are very useful in further reduction of the false alarm ratio. A comprehensive evaluation showed that the system has high accuracy of fall detection and very low level of false alarms. It demonstrated that the placement of the cameras does not have an influence on the classification accuracy.

The depth images were acquired by the Kinect sensors using OpenNI. The system was implemented in C/C++ and runs at 25 fps on 2.4 GHz I7 notebook. The most computationally demanding operation is extraction of the depth reference image of the scene. For images of size 640×480 the computation time needed for extraction of the depth reference image is about 9 milliseconds. In order to reduce the computational overload the depth reference images were only updated if on the image acquired in the moment of the fall, two or more blobs had been detected. In practice, we examined the thresholded difference between the current depth map and the reference map in terms of the number of blobs.

7 Conclusions

In this work we demonstrated how to achieve reliable fall detection with low false positives number. Given the alarm trigger obtained on the basis of data from wireless accelerometer, the system extracts the person features from the corresponding depth image and point clouds. The system uses them in a k-NN classifier to examine if the person is lying on the floor. In order to further reduce the false alarm ratio the system extracts fall energy images from a sequence of images up to the fall and then employs the energy features in a k-NN classifier. Experimental results demonstrated that this leads to considerable reduction of false alarms and high detection ratio. The system preserves the privacy of the user and works in poor lighting conditions.

Acknowledgment. This work has been supported by the National Science Centre (NCN) within the project N N516 483240.

References

1. Ahad, M.A.R., Tan, J.K., Kim, H., Ishikawa, S.: Motion history image: its variants and applications. *Mach. Vision Appl.* 23(2), 255–281 (2012)
2. Bourke, A., O'Brien, J., Lyons, G.: Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & Posture* 26(2), 194–199 (2007)
3. Cleary, J., Trigg, L.: An instance-based learner using an entropic distance measure. In: *Int. Conf. on Machine Learning*, pp. 108–114 (1995)
4. Cover, T.M., Thomas, J.A.: *Elements of Information Theory*. Wiley (1992)
5. Cover, T.M., Thomas, J.A.: *Data Mining: Practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)
6. Cucchiara, R., Prati, A., Vezzani, R.: A multi-camera vision system for fall detection and alarm generation. *Expert Systems* 24(5), 334–345 (2007)
7. Foroughi, H., Naseri, A., Saberi, A., Yazdi, H.: An eigenspace-based approach for human fall detection using integrated time motion image and neural network. In: *9th Int. Conf. on Signal Processing*, pp. 1499–1503 (2008)
8. Horn, B.: *Robot Vision*. The MIT Press, Cambridge (1986)
9. Jansen, B., Deklerck, R.: Context aware inactivity recognition for visual fall detection. In: *Proc. IEEE Pervasive Health Conference and Workshops*, pp. 1–4 (2006)
10. Kepski, M., Kwolek, B., Austvoll, I.: Fuzzy inference-based reliable fall detection using Kinect and accelerometer. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *ICAISC 2012, Part I. LNCS*, vol. 7267, pp. 266–273. Springer, Heidelberg (2012)
11. Kepski, M., Kwolek, B.: Human fall detection using Kinect sensor. In: Burduk, R., Jackowski, K., Kurzynski, M., Wozniak, M., Zolnierrek, A. (eds.) *CORES 2013. AISC*, vol. 226, pp. 743–752. Springer, Heidelberg (2013)
12. Labayrade, R., Aubert, D., Tarel, J.P.: Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In: *Intelligent Vehicle Symposium*, vol. 2, pp. 646–651. IEEE (June 2002)
13. Marshall, S.W., Runyan, C.W., Yang, J., Coyne-Beasley, T., Waller, A.E., Johnson, R.M., Perkis, D.: Prevalence of selected risk and protective factors for falls in the home. *American J. of Preventive Medicine* 8(1), 95–101 (2005)
14. Mastorakis, G., Makris, D.: Fall detection system using Kinect's infrared sensor. *J. of Real-Time Image Processing*, 1–12 (2012)
15. Miaou, S.G., Sung, P.H., Huang, C.Y.: A customized human fall detection system using omni-camera images and personal information. *Distributed Diagnosis and Home Healthcare*, 39–42 (2006)
16. Mubashir, M., Shao, L., Seed, L.: A survey on fall detection: Principles and approaches. *Neurocomputing* 100, 144–152 (2013), special issue: Behaviours in video
17. Noury, N., Fleury, A., Rumeau, P., Bourke, A., ÓLaighin, G., Rialle, V., Lundy, J.: Fall detection - principles and methods. In: *Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, pp. 1663–1666 (2007)
18. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Monocular 3D head tracking to detect falls of elderly people. In: *Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, pp. 6384–6387 (2006)
19. Shi, G., Chan, C.S., Li, W.J., Leung, K.S., Zou, Y., Jin, Y.: Mobile human airbag system for fall protection using MEMS sensors and embedded SVM classifier. *IEEE Sensors Journal* 9(5), 495–503 (2009)

Modified Dendrite Morphological Neural Network Applied to 3D Object Recognition on RGB-D Data

Humberto Sossa and Elizabeth Guevara

Instituto Politécnico Nacional, CIC. Av. Juan de Dios Batiz S/N,
México, D.F. C.P. 077600
hsossa@cic.ipn.mx

Abstract. In this paper a modified dendrite morphological neural network (DMNN) is applied for 3D object recognition. For feature extraction, shape and color information were used. The first two Hu's moment invariants are calculated based on 2D grayscale images, and color attributes were obtained converting the RGB (Red, Green, Blue) image to the HSI (Hue, Saturation, Intensity) color space. For testing, a controlled lab color image database and a real image dataset were considered. The problem with the real image dataset, without controlling light conditions, is that objects are difficult to segment using only color information; for tackling this problem the Depth data provided by the Microsoft Kinect for Windows sensor was used. A comparative analysis of the proposed method with a MLP (Multilayer Perceptron) and SVM (Support Vector Machine) is presented and the results reveal the advantages of the modified DMNN.

Keywords: Dendrite morphological neural network, 3D object recognition, Kinect, depth segmentation, color, classification.

1 Introduction

Object recognition is an important task in computer vision due to its variety of applications in many areas of artificial intelligence including, for example, content-based image retrieval, industrial automation or object identification for robots [15].

A visual system recognizes objects by multiple features, including shape, color and texture. In this case, the shape and the color of the object are the features selected for representation. For shape characterization, the Hu's moment invariants are calculated and for color features, the HSI color space is used as an alternative to the RGB space because this model is suitable for processing images that present lighting changes.

On the other hand, image segmentation is very critical to image processing for object location. The existing algorithms that use only the color information for the segmentation have some difficulties. Usually the objects to segment must have a big contrast with background to easily distinguish the object. To tackle

this problem, the depth data from the Kinect sensor is used. In this way, the segmentation problem of the objects in real scenarios becomes simpler because the segment boundary can be defined like the edges on depth data.

This paper describes a method to recognize 3D objects through a pattern recognition approach using a modified DMNN for classification [13]. The proposed method of object recognition system has two phases: training and testing phase. During the training phase, the images are preprocessed and the feature vector is generated. The feature vector is stored with the object label and the DMNN is trained. During testing phase, the test image is given to the system; the features are extracted of the preprocessed image and then the classifier is employed to recognize the object.

The system was evaluated on 20 objects taken from the COIL-100 color database [8] and on a set of images captured without controlled lighting condition, the depth object segmentation was used in this last dataset.

The rest of the paper is organized as follows. Section 2 describes the feature extraction process and Section 3 presents the depth segmentation procedure used in the real dataset. Section 4 explains the modified DMNN. Section 5 presents the proposed object recognition method. Section 6 is focused to present the experimental results where the classification method used in the recognition process is tested and compared with other classifiers. Finally, Section 7 is oriented to provide the conclusions and directions for further research.

2 Feature Extraction Process

Feature extraction is a process for converting the input data into a set of features that extract the relevant information in order to perform a task, in this case, recognize an object. In this paper, we use Hu's moment invariants and color to represent the object.

2.1 Moment Invariants

Geometric moment invariants were introduced by Hu based on the theory of algebraic invariants [6]. Image or shape feature invariants remain unchanged if the image undergoes any combination of the following transformations: translation, rotation and scaling. Therefore, the moment invariants can be used to recognize the object even if the object has changed in certain transformations.

A set of seven invariant moments can be derived from the second and third moments, in this paper we only use the first and second invariant moments to represent the object. Details can be found in [4].

2.2 Color Features

Color spaces provide a method for specifying, ordering and manipulating colors. In this paper, the HSI color space is used as an alternative to RGB space because HSI is less sensitive to the lighting changes than the RGB [7]. The HSI space

considers the image as a combination of the components: hue (H), saturation (S) and intensity (I).

The RGB components of an image can be converted to HSI color representation by:

$$H = \cos^{-1} \left(0.5 (R - G) + (R - B) / \sqrt{(R - G)^2 + (R - B)(G - B)} \right). \quad (1)$$

In order to have the value for hue in the range from 0 to 360 degrees, it is necessary to subtract H from 360° when $B > G$. The formulas for saturation and intensity are, respectively:

$$S = 1 - \frac{3}{R + G + B} \min(R, G, B), \quad I = \frac{1}{3} (R + G + B). \quad (2)$$

Generally, hue is considered as an angle between a reference line and the color point in RGB space; the range of the hue value is from 0 to 360 degrees. When the saturation component is close to 0, colors only reflect a change between black and white. When this component is close to 1, the color will reflect the true value represented by the hue. To determine the colors black, white and gray, the saturation component is used, if saturation is close to 0 then whether the intensity is close to 0 the color is black, if it is close to 1 the color is white and otherwise, the color is gray.

3 Segmentation Process

Image segmentation is the task of partitioning an image into consistent regions and is typically used to identify objects or other relevant information in digital images. The existing algorithms that use only the color information for the segmentation have some difficulties. To obtain a good segmentation, usually the objects to segment must have a big contrast with background to make it easier to distinguish the object. It is also important to have a homogeneous background to avoid that wrong areas are marked as objects; and the illumination circumstances must be constants. To tackle this problem, the depth data provided by the Kinect sensor is used. In this way, the segmentation problem of the objects in real scenarios becomes simpler because the segment boundary can be defined like the edges on depth data.

For segmentation is very important that the depth and image cameras on the Kinect are aligned because the detection of the object in the depth image (their contour) must coincide with the color image to achieve a good segmentation. To calibrate the two cameras, the calibration tool reported in [1] was used.

On the other hand, the depth map of the Kinect for Windows in the near mode has an effective range of approximately 0.5 to 3 meters. In this paper, the depth information in the raw depth per pixel was used to divide the depth image in layers where only the first layer was taking into account, this layer goes from

0.5 to 0.9 meters and only the object that is allocated to this distance from the Kinect is considered for recognition.

For explaining the segmentation process, an object from the real dataset was selected as an example. Figure 1a) presents the interest object in the scene, applying the depth division, the regions recognized by the Kinect in the first layer are presented in Figure 1b), as can be seen the object was correctly segmented. However, the depth data has noise and holes; especially, in the silhouettes of the objects, these problems are recurring; furthermore, the Kinect detects points from the surface too. To reduce these problems, the color image was binarized (Fig. 1c)) and the noise was eliminated using erosion and dilation filters; Fig. 1d) presents the filtered image and Fig. 1e) presents the corresponding color image. To get a noiseless image for extracting the features representation, the minimum and maximum points in x and y were determined from the binary filtered image and a rectangular area was calculated; this area was selected from the original color image and the result for this example is presented in Fig. 1f).



Fig. 1. a) Scenario for image capture, b) Depth segmentation, c) Binary segmented image, d) Filtered segmented image, e) Color segmented image, f) Test image

4 Dendrite Morphological Neural Networks

The dendrite morphological neural networks (DMNN) were first described by Ritter and colleagues in [9] and [11]. DMNN emerge as an improvement of classical morphological neural networks (MNN), originally introduced by Davidson in [3] and then re-discussed by Ritter and Sussner in [10]. A key issue in the design of a DMNN is its training; this is in the selection of the number of dendrites and the values of synaptic weights for each dendrite. Diverse algorithms to automatically train a DMNN can be found in [9], [12] and [2]. A novel algorithm for the automatic training of a DMNN was proposed in [13] and it is applied in this work for object recognition.

4.1 Basics on Dendrite Morphological Neural Networks

Morphological neural networks are closely related to Lattice Computing [5], which can be defined as the collection of Computational Intelligence tools and techniques that use lattice operations \vee (maximum), or \wedge (minimum), and $+$

from the semirings $(\mathbb{R}_{-\infty}, \vee, +)$ or $(\mathbb{R}_{\infty}, \wedge, +)$ where $\mathbb{R}_{-\infty} = \mathbb{R} \cup \{-\infty\}$ and $\mathbb{R}_{\infty} = \mathbb{R} \cup \{\infty\}$. From a set of n input neurons, the computation at a neuron in a MNN for input $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is given by

$$\tau_j(\mathbf{x}) = a_j \bigvee_{i=1}^n b_{ij}(x_i + w_{ij}) \quad \text{or} \quad \tau_j(\mathbf{x}) = a_j \bigwedge_{i=1}^n b_{ij}(x_i + w_{ij}) \quad (3)$$

where $b_{ij} = \pm 1$ denotes if the i th neuron causes excitation or inhibition on the j th neuron, $a_j = \pm 1$ denotes the output response (excitation or inhibition) of the j th neuron to the neurons whose axons contact the j th neuron and w_{ij} denotes the synaptic strength between the i th neuron and the j th neuron. Parameters b_{ij} and a_j take +1 or -1 value if the i th input neuron causes excitation or inhibition to the j th neuron. The computation performed by the k th dendrite (D_k) can be expressed by the formula:

$$D_k(\mathbf{x}) = a_k \bigwedge_{i \in I} \bigwedge_{l \in L} (-1)^{1-l} (x_i + w_{i_k}^l) \quad (4)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ corresponds to the input neurons, $I \subseteq \{1, \dots, n\}$ denotes to the set of all input neurons N_i with terminal fibers that synapse on the k th dendrite of a morphological neuron N , $L \subseteq \{0, 1\}$ corresponds to the set of terminal fibers of the i th neuron that synapse on the k th dendrite of N , and $a_k \in \{-1, 1\}$ denotes the excitatory or inhibitory response of the k th dendrite.

The activation function used in a MNN is the hard limiter function. A more detailed explanation can be found in [9] and [12].

4.2 The Training Algorithm

The proposed training algorithm for a DMNN in [13] is summarized below. Given p classes of patterns, C^k , $k = 1, 2, \dots, p$, each with n attributes, the algorithm applies the following steps:

Step 1) Select the patterns of all the classes and open a hyper-cube HC^n (with n the number of attributes) with a size such that all the elements of the classes remain inside HC^n . The hyper-cube can be one whose coordinates match the patterns of class boundaries; it can be called the minimum hyper-cube MHC . For having better tolerance to noise at the time of classification, add a margin M on each side of the MHC . This margin is a number greater or equal to zero and is estimated as a function of the size T of the MHC . If $M = 0.1T$ then the new hyper-cube will extend that ratio to the 2^n sides of the MHC .

Step 2) Divide the global hyper-cube into 2^n smaller hyper-cubes. Verify if each generated hyper-cube encloses patterns from only one class. If this is the case, label the hyper-cube with the name of the corresponding class, stop the learning process and proceed to step 4.

Step 3) If at least one of the generated hyper-cubes (HC^n) has patterns of more than one class, then divide HC^n into 2^n smaller hyper-cubes. Iteratively repeat the verification process onto each smaller hyper-cube until the stopping criterion is satisfied.

Step 4) Based on the coordinates on each axis, calculate the weights for each hyper-cube that encloses patterns belonging to C^k . By taking into account only those hyper-cubes that enclose items C^k and at this moment the DMNN is designed. Figure 2 shows a dendrite morphological neural network with an input layer that separates the two classes: C^1 and C^2 . The neurons of the input layer are connected to the next layer via the dendrites. The black and white circles denote excitatory and inhibitory connection respectively. The geometrical interpretation of the computation performed by a dendrite is that every single dendrite defines a hyperbox which can be defined by a single dendrite via its weight values w_{ij} as the example shows.

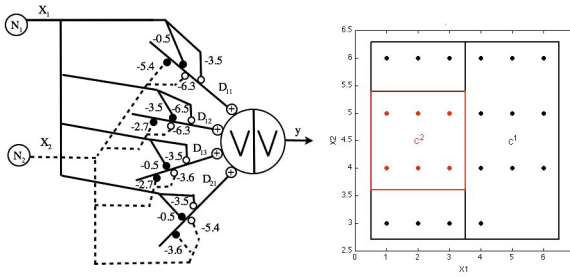


Fig. 2. Morphological neural network to solve the problem that appears on the right side of the figure. Points of class C^1 (black solid dots) are enclosed by the three black boxes and C^2 (red solid points) by the red box generated by the dendrites.

5 Object Recognition System Overview

The flowchart of the framework for object recognition is shown in Fig. 3. For recognition purpose, a modified DMNN is used as classifier supported by features extracted from the given images. Hu’s moment invariants and color information in HSI scale are the features that represent the object. In the case of the Kinect dataset, the preprocessing includes the depth segmentation algorithm explained in Section 3.

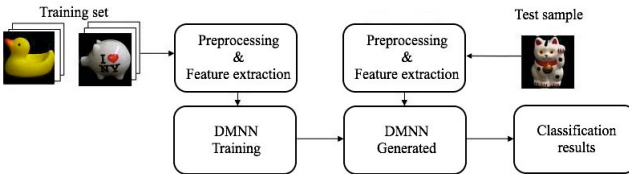


Fig. 3. Object recognition process

6 Experimental Results and Discussion

To test the performance of the object recognition approach, we used a set of RGB-D images captured without controlled lighting condition and a subset of objects from the COIL-100 dataset. The COIL-100 dataset [8] consists of color images of 100 different objects with black background, each one is rotated with 5 degree angle interval in vertical axis. Hence for every object there are 72 images. The size of the images is of 128x128 pixels. The 20 objects subset selected from the COIL-100 is shown in Fig. 4, these objects correspond to the images from the COIL-20 dataset (set of 20 gray images) but in color.



Fig. 4. Subset of COIL-100 database

The experimental dataset consists of 10 objects with 10 images each one. The images were captured using the Kinect for Windows RGB camera with a resolution of 640x480 and the depth image was captured with the same resolution in the near mode of the Kinect. Figure 5 a) shows the RGB images from the dataset objects. As can be seen some of these objects are very similar in shape and therefore, the color is important to distinguish between them.

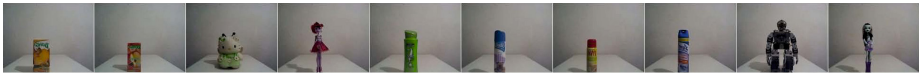


Fig. 5. Real dataset of 10 objects

Figure 1a) presents an example of the real background where the images were captured. It is obvious that these images were not taken under strict illumination controlled condition and that there are other objects besides the interest object; in spite of this, results were satisfactory. Figure 6 presents 4 examples of the ten views used in the real dataset from one of the objects, the images show the segmented object from each scene obtained applying the depth segmentation process explained in Section 3.

For object representation, the first and second Hu's moments obtained from gray images were used and for the color, the system detects 8 colors (red, yellow, green, blue, magenta, white, gray and black) from the HSI color scale. The color features were represented as the percentage of each color in the object. Therefore, the objects were characterized by 10 features (8 of color and 2 of shape).



Fig. 6. Sample images of one object

All the algorithms were implemented on a desktop computer with Intel i7 2.2 GHz processor with 8GB in RAM. The preprocessing, feature extraction and the DMNN were implemented in MATLAB 7.11. To evaluate the performance, the DMNN was compared with a Multilayer Perceptron (MLP) and Support Vector Machine (SVM). The MLP and SVM were applied using the software Weka 3-6-8. For the MLP with one hidden layer, the training parameters were established as: learning rate=0.3 and momentum=0.2, the activation function used was the sigmoid. For the SVM we used a Polynomial Kernel of degree 2.

As a first experiment, we used the images at 0, 45, 90, 135, 180, 225, 270 and 315 degrees for training on each of the 20 classes of the COIL-100 subset. Testing was done on the remaining images. Table 1 shows that the error obtained with the modified DMNN is better than the error obtained with the MLP and the SVM.

Table 1. Comparison table of the MLP, SVM and modified DMNN for the subset of COIL-100 dataset

MLP		SVM		DMNN	
# Neurons	% of Recognition	% of Recognition	# Dendrites	M	% of Recognition
34	87.29	86.67	66	0.481	87.79

In a previous work [14], this experiment was performed without considering the color information, only the mean and the standard deviation of the distribution of the pixels in the gray images and the two Hu's moment invariant were calculated. For the COIL-20 dataset, the percentage of recognition was of 82.27% which was improved to 87.79% using color information. These results reveal the good performance of the modified DMNN for object recognition and that the color characteristics help to improve the percentage of recognition.

To have an estimated error generalization, 10 experiments were performed with training and testing samples randomly selected for both datasets. For the COIL-100 subset, 8 images of each object were randomly selected (160 samples) for training and the other 1280 samples were used for testing. Table 2 shows the average of the percentage of recognition and the standard deviation. These results reveal that the performance of the modified DMNN for object recognition improves the results obtained with the MLP and SVM; furthermore the standard deviation of DMNN results is smaller than the standard deviation of the MLP and the SVM, so the performance of the DMNN is satisfactory.

For the real dataset, the system was trained with 3 random images of each object and the rest of the images (7) were used for testing. Table 3 presents the

Table 2. Comparison table of the MLP, SVM and modified DMNN for the subset of COIL-100 dataset

	MLP	SVM	DMNN
Average % of Recognition	84.99	83.14	86.18
Standard deviation	1.57	1.70	1.53

average of the percentage of recognition for the 10 objects dataset and the standard deviation obtained with the 10 experiments. As can be seen, the modified DMNN and the SVM have the same percentage of recognition and it improves the result of the MLP, so the performance of the proposed algorithm [13] is satisfactory for object recognition. Also, in this experiment, the standard deviation of DMNN is smaller than the standard deviation of the SVM and MLP, which is a good characteristic because the percentage of recognition is more constant for different training sets.

Table 3. Comparison table of the MLP, SVM and modified DMNN for the real dataset

	MLP	SVM	DMNN
Average % of Recognition	80.71	81.85	81.85
Standard deviation	5.17	7.01	4.96

Selecting specific views of the objects, the percentage of recognition improved to 90% for the DMNN and SVM and 87.15% for the MLP; hence, the results in both experiments show that it is convenient to include images of specific views of the object in the training dataset to improve the recognition percentage. With respect to training computation time, the DMNN requires more time than the SVM and MLP, however the DMNN algorithm can be implemented on a parallel architecture which would improve this point.

7 Conclusions

In this paper a modified DMNN was applied for 3D object recognition. By using 2D moments and reduced color information, the proposed method does not require complex features calculation, thus reducing process time in the feature extraction stage.

The results achieved suggest that segmentation using depth information could be a useful tool, however further research and experimentation is needed to verify this idea. Comparisons of the modified DMNN with the MLP and SVM demonstrated that the DMNN can be applied to solve the recognition problem. The simplicity of the extracted features and the DMNN calculation allow that the proposed recognition method can be used in real applications and future work will be focused on this way. Furthermore, the implementation of DMNN training

algorithm on a parallel architecture is necessary for evaluating the method in larger databases with more describing features and for improving the training computation time.

Acknowledgments. H. Sossa would like to thank SIP-IPN and CONACYT under grants 20121311, 20131182 and 155014 for the economical support to carry out this research. E. Guevara thanks CONACYT for the scholarship granted to pursue her doctoral studies.

References

1. Burrus, N.: Kinect RGB Demo (2011)
2. Chyzhyk, D., Graña, M.: Optimal hyperbox shrinking in dendritic computing applied to Alzheimer's disease detection in MRI. In: Corchado, E., Snásel, V., Sedano, J., Hassanién, A.E., Calvo, J.L., Ślęzak, D. (eds.) SOCO 2011. AISC, vol. 87, pp. 543–550. Springer, Heidelberg (2011)
3. Davidson, J.L., Hummer, F.: Morphology neural networks: An introduction with applications. *Circuits Systems Signal Process* 12(2), 177–210 (1993)
4. González, R., Woods, R.: *Digital Image Processing*. Pearson (2007)
5. Graña, M.: Special issue on: Lattice computing and natural computing. *Neurocomputing* 72(10-12), 2065–2066 (2009)
6. Hu, M.K.: Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory* 8, 179–187 (1962)
7. Jain, R., Kasturi, R., Schunck, B.G.: *Machine Vision*. McGraw-Hill (1995)
8. Nene, D., Nayar, S., Murase, H.: Columbia object image library: COIL-100 (1996)
9. Ritter, G.X., Iancu, L., Urcid, G.: Morphological perceptrons with dendritic structure. In: 12th IEEE International Conference in Fuzzy Systems, FUZZ 2003, vol. 2, pp. 1296–1301 (2003)
10. Ritter, G.X., Sussner, P.: An introduction to morphological neural networks. In: *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 4, pp. 709–717 (1996)
11. Ritter, G.X., Urcid, G.: Lattice algebra approach to single-neuron computation. *IEEE Transactions on Neural Networks* 14(2), 282–295 (2003)
12. Ritter, G.X., Urcid, G.: Learning in lattice neural networks that employ dendritic computing. *Computational Intelligence Based on Lattice Theory* 67, 25–44 (2007)
13. Sossa, H., Guevara, E.: Efficient training for dendrite morphological neural networks. Submitted to *Neurocomputing - Elsevier Journal*
14. Sossa, H., Guevara, E.: Modified dendrite morphological neural network applied to 3D object recognition. In: Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Rodríguez, J.S., di Baja, G.S. (eds.) MCPR 2012. LNCS, vol. 7914, pp. 314–324. Springer, Heidelberg (2013)
15. Wöhler, C.: *3D Computer Vision: Efficient Methods and Applications*. Springer (2012)

Diversity Measures for Majority Voting in the Spatial Domain

Andras Hajdu, Lajos Hajdu, Laszlo Kovacs, and Henrietta Toman

Faculty of Informatics, University of Debrecen
Egyetem ter 1, 4010 Debrecen POB 12, Hungary
hajdu.andras@inf.unideb.hu, hajdul@math.klte.hu,
{kovacs.laszlo.ipgd,toman.henrietta}@inf.unideb.hu

Abstract. The classic majority voting model can be extended to the spatial domain e.g. to solve object detection problems. However, the detector algorithms cannot be considered as independent classifiers, so a good ensemble cannot be composed by simply selecting the individually most accurate members. In classic theory, diversity measures are recommended that may help to explore the dependencies among the classifiers. In this paper, we generalize the classic diversity measures for the spatial domain within a majority voting framework. We show that these measures fit better to spatial applications with a specific example on object detection on retinal images. Moreover, we show how a more efficient descriptor can be found in terms of a weighted combination of diversity measures which correlates better with the accuracy of the ensemble.

Keywords: classifier combination, majority voting, spatial domain, diversity measures, biomedical imaging.

1 Introduction

In decision making, the accuracy of the decision can be increased by composing ensembles from individual classifiers. In our previous work [1], we generalized the classical majority voting model to be applicable in the spatial domain. Namely, we introduced the terms $0 \leq p_{n,k} \leq 1$ describing the probability of a correct decision if k correct votes are present among the total number of n . This generalization was motivated by object detection problems in digital images, where image processing algorithms (detectors) are the members of the ensemble. Each individual algorithm votes in terms of a single pixel/region as its candidate for the center/object in the image domain. The region matching the geometry of the object with maximal number of candidates included is considered as the decision. Only the votes falling inside a proper region can vote together for the object. A good decision can be made even if the false candidates have majority, while bad decision is made only when a subset of false candidates with larger cardinality than the number of correct ones can be covered by a region matching the geometry of the object.

In classic majority voting, only the correctness of the votes influences the decision. However, in the object detection scenario, the spatial behavior of the

votes are also important. Majority voting can be applied in the generalized model with further geometrical constraints (e.g. the spatial closeness of the candidates) that can be described by the terms $p_{n,k}$.

We applied the generalized model for the detection of the optic disc (OD) in retinal images. The OD is a bright region with circular shape having diameter d_{OD} (clinically predetermined constant). For the output of each detector for the OD center we consider the minimal bounding circles for all subgroups of the candidates. The circle with maximal number of candidates, having diameter less than or equal to d_{OD} is chosen for the OD as it is illustrated in Fig. 1.

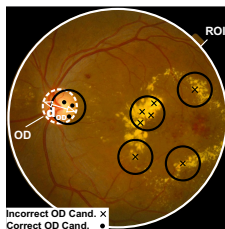


Fig. 1. Optic disc detection using spatial majority voting, where the black circles show the possible hotspots containing different number of OD candidates. The black dots and the black crosses represent the true and false OD candidates respectively.

In our application, the participating OD detector algorithms have individual respective accuracies $p_1 = 0,6472$; $p_2 = 0,9765$; $p_3 = 0,3205$; $p_4 = 0,7593$; $p_5 = 0,3153$; $p_6 = 0,2276$; $p_7 = 0,9582$; $p_8 = 0,7671$, [2] on the Messidor dataset [3] containing 1200 retinal images with resolution 2240*1488 pixels. All the quantitative results presented later in the paper correspond to these.

In the literature of classic majority voting, several results are achieved for independent voters, but in object detection problems, the individual algorithms can hardly be expected to be independent. Besides the individual accuracies of the detector algorithms, the dependencies among them should also be taken into consideration, when an ensemble is composed from them.

In decision making theory, a possible simple approach to estimate dependency of the members is to consider diversity measures. These measures are defined between classifiers in [4]. In [5], it is proposed that we can reach optimally performing classifier combination by making up classifiers with high individual accuracies and sufficient level of diversity at the same time. Several earlier works (e.g. [6,8]) confirmed that neither individual performances nor diversity alone can guarantee that the ensemble outperforms all the individual classifiers. Recent works (e.g.[4]) have been focused on finding suitable diversity measures, when majority voting is considered as a decision rule. Our motivation is to check the reliability of these diversity measures in the spatial domain and to generalize them for better performance. The generalization is done in a natural way: we follow similar principles here that are considered also in the generalization of majority voting to the spatial domain.

The rest of the paper is organized as follows. In section 2 we list the diversity measures recommended in classic theory. Section 3 is dedicated to the generalization of the classic diversity measures. In section 4, we compare the performance of the classic and generalized measures in our spatial application. Section 5 introduces a novel methodology to derive a combined diversity measures from the individual ones. Finally, in section 6, we draw some conclusions.

2 Diversity Measures in Classic Voting Theory

Depending on whether it assesses the pairwise or groupwise dissimilarity, two types of diversity measures are considered. If a system of M classifiers $D = \{D_1, \dots, D_M\}$ is given, let y_{ij} denote the classifier output of the j -th classifier for the i -th input sample. Let $\mathbf{y}_i = [y_{i1}, \dots, y_{iM}]^T$ denote the joint output of a system for the i -th input sample x_i . Assuming that the output has binary form, $y_{ij} = 1$ means correct, while $y_{ij} = 0$ means incorrect classification. As the measures are mainly based on simple binary algebra, the following simplifications can be introduced, if we compare two classifiers with a diversity measure. Let N^{ab} , $a, b \in \{0, 1, *\}$ denote the number of input samples, where $*$ stands for any of the output 0 or 1. Here a belongs to the first classifier and b to the second one; i.e. N^{ab} and N^{ba} are different. The number of classifiers producing error on the input sample x_i ($i = 1, \dots, n$) is denoted by $m(x_i)$ which can be expressed as $m(x_i) = \sum_{j=1}^M (1 - y_{ij})$. Finally the error rate of the j -th classifier can be calculated as $e_j = \frac{1}{N} \sum_{i=1}^N (1 - y_{ij})$.

In the literature (see [4,8]) the following diversity measures are defined: minimum individual error, mean error, majority voting error, majority voting improvement, correlation coefficient, product-moment correlation measure, Q-statistics, disagreement measure, double-fault measure, entropy measure, measure of difficulty, Kohavi-Wolpert variance, interrater agreement measure, fault majority measure. Now we give a brief overview of some diversity measures from those can be considered for generalization to our spatial model.

- *The correlation coefficient C2*: it is a well known and frequently used statistical measure. For binary classifier output its definition takes the form:

$$C2_{ij} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{N^{1*}N^{0*}N^{*1}N^{*0}}}.$$

- *The disagreement measure D2*: it depends on the number of samples for which the classifiers disagreed and the total number of observations. It is calculated as:

$$D2_{ij} = \frac{N^{01} + N^{10}}{N}.$$

- *Mean error ME*: this measure takes the average of individual classifier error rates within the ensemble and can be defined by the following formula:

$$\bar{e} = \frac{1}{M} \sum_{i=1}^M e_j.$$

- *Interrater agreement measure IA*: this measure characterizes the level of agreement. With the notation presented above it can be expressed as:

$$IA = 1 - \frac{\sum_{i=1}^N m(x_i)(M - m(x_i))}{NM(M - 1)\bar{e}(1 - \bar{e})}.$$

3 Generalized Diversity Measures for the Spatial Domain

The diversity measures in section 2 give useful information on how to select the members to achieve the highest ensemble performance. More specifically, we can consider the correlation between the diversity measures and the system accuracy. In the literature, the case when the classifier decision making method is not the majority rule is rarely examined. These measures can be modified to the non-majority voting case in the spatial domain. In many image processing applications, more algorithms are used to detect the same object in the image. These algorithms can be considered as classifiers in a fusion method and the output of the algorithms (pixels/regions) as votes. In this voting method the good votes have to fulfill some geometrical constraints. Good decision can be made even in that case when the number of good votes is less than the half of the total votes. To achieve the best performance, the algorithms not making coincident error have to be combined. It can be proved that the modified diversity measures for the non-majority case can provide the possibility for better classifier selection. The aim of modifying the diversity measures is to reach higher correlation between them and the system accuracy. We could modify the calculation of the classic measures so that the original coherence in our specific environment is described. In this way, the generalized diversity measures consider the geometrical constraints, adopting them with getting close to each other.

This modification is logical, since close votes outside the good area cause the main problem. Some other interpretations of the pairwise diversity measures were investigated, as well. In some cases all, the variants correlated more with the system accuracy than the original diversity measure. The following formulas correlated most with the system accuracy are introduced here as generalized diversity measures for the spatial domain:

- *The generalized correlation coefficient C2'*:

$$C2'_{ij} = \frac{N^{11}N^{0'0'} - N^{01}N^{10}}{\sqrt{N^{1*}N^{0*}N^{*0}}}.$$

To handle spatial behavior of votes, now we consider also the notation $N^{0'0'}$. This figure stands for the number of cases, where for a pair of classifiers both of them made bad decision and these votes also fulfill the geometric constraint (that is, close to each other). Similarly, N^{00} means that though both algorithms give bad candidates, it does not mean a problem, because the geometrical constraints are not satisfied, so the chance for a final bad decision is not increased.

The modification of the other diversity measures, defined between two classifiers, can be interpreted in the same way. For the disagreement measure and that numbers describing the whole system of classifiers, (e.g. the interrater agreement measure), the generalization for our model needs some further modifications.

- *The generalized disagreement measure $D2'$* : it depends on the number of samples for which the classifiers disagreed and the total number of observations. In this case all possible disagreement situations have to be described in the modified formula. It can be written as:

$$D2'_{ij} = \frac{N^{01} + N^{10} + N^{0'1} + N^{10'} + N^{0'0} + N^{00'}}{N},$$

where for example $N^{0'1}$ describes the number of the situations where one of the classifiers give bad vote fulfilling the geometrical constraints and the other give good vote.

- *The interrater agreement measure IA'* : this measure characterizes the level of agreement. With the notation presented above it can be expressed as:

$$IA' = 1 - \frac{\sum_{i=1}^N m'(x_i)(M - m'(x_i))}{NM(M - 1)\bar{e}(1 - \bar{e})}.$$

In the classic formula $m(x_i)$ is the number of classifiers producing error for the input sample, and $m'(x_i)$ expresses the number of bad votes which are relevant in making the final decision, so the bad candidates fulfill the geometrical constraints, as well.

The plots in Fig. 2 (a), (b), (c) and (d) show examples about the effectiveness of the generalized diversity measures. Each dashed line shows the correlation between the system accuracy and the modified diversity measures. It can be observed that after modification this correlation is increased for each diversity measure.

Another interesting fact is that in the spatial domain we can handle ensembles consisting of an even number of voters, as well. Namely, in most of the classic studies the results are presented only for odd number of classifiers. The reason is that in classical majority voting, adding a new classifier can drop the system accuracy, so we cannot guarantee to achieve better performance because the parity of the number of the classifiers changes. This phenomena can be observed by the correlation curve of the diversity measures, as well.

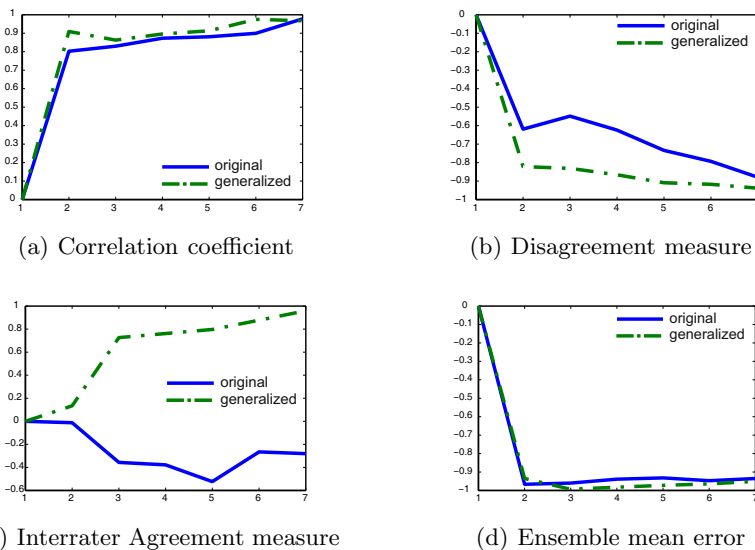


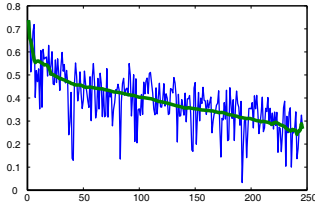
Fig. 2. Comparison of the generalized (dashed line) and the original (solid line) pairwise (a),(b) and non-pairwise (c),(d) diversity measures. The x-axis represents the number of the classifiers in the ensemble, while the y-axis the correlation value. The higher the absolute value of the correlation is, the better the effectiveness of the diversity measures is.

4 Distortion of the Ensemble Members

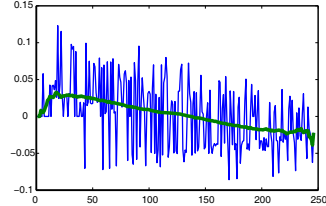
For generating the most accurate ensemble, the distortion of the algorithms is a relevant issue for applications. The distortion can be described as the difference between the optimal (real), and the actual output of the algorithms. If in such cases, the reason or the magnitude of the distortion is known, the inverse distortion vector can be calculated. By the help of this vector, the deviation can be reduced and the actual output can get closer to the optimal (real) value. The diversity measures can be built in our generalized model which is used for optic disc detection as an application. Using the inverse distortion, the achieved performance of the ensemble system is relevantly higher than the original (distorted) one. In this section we show, that for the diversity measures the inverse distortion step cannot be ignored.

The main problem with the diversity measures for a majority voting system is the amount of available training data. The high performance of the ensemble-based system generates few amount of data regarding bad votes, but the most of the diversity measures are built upon this information. For instance, the most important situation for our application is when the bad votes fulfilling the geometrical constraints may cause wrong final decision. Without appropriate number of such situations, the diversity measures generate incorrect values, which results in high distortion and low correlation with the system accuracy. By low

correlation, the recommendation for the ensemble system is not satisfied. While the main motivation of the usage of diversity measures is to find the system with the best performance, sufficient number of special situations is not available, but can be interpolated. In our proposed model and in the application, all the diversity measures are smoothed to suppress the lack of data. Fig. 3 (a), (b) show the result of the smoothing step. This step is required not just for our modified diversity measure, but for the original ones, as well.



(a) Ensemble mean error interpolation for the original diversity measure.



(b) Disagreement measure interpolation for the modified diversity measure.

Fig. 3. Comparison of the diversity measures before and after interpolating the missing cases. After interpolation, the curves are strongly smoothed i.e. abnormal values are removed. The x-axis represents the number of combinations of classifiers (247 different situations exist regarding 8 classifiers, where the diversity can be measured), while the y-axis the value of the diversity measure.

Fig. 4 (a) and (b) show, that after the interpolation step, the correlation between the system accuracy and the diversity measures is increased dramatically in both cases. The dashed lines show that after applying the interpolation, the correlation values are considerably increased independently whether a modification was applied or not. In case of non-pairwise diversity measure, Fig. 4 (c), and (d) show the similar results as Fig. 4 (a), and (b).

5 A Weighted Combination of Diversity Measures

While in most cases the dependencies between the assembled classifiers are unknown (e.g. between the algorithms in our OD application), by generating an ensemble from the classifiers having the highest accuracies the optimal performance may not be achieved. Although the diversity measures suggested by the literature are extended successfully in section 3, and their performance is improved by applying the interpolation, it cannot be guaranteed to choose the ensemble having the best accuracy regarding diversity measures. For solving this problem the diversity measures can be considered as feature selectors and a weighted linear combination scheme can be applied for them. That is suppose that M classifiers and I diversity measures are given and the aim is to compose

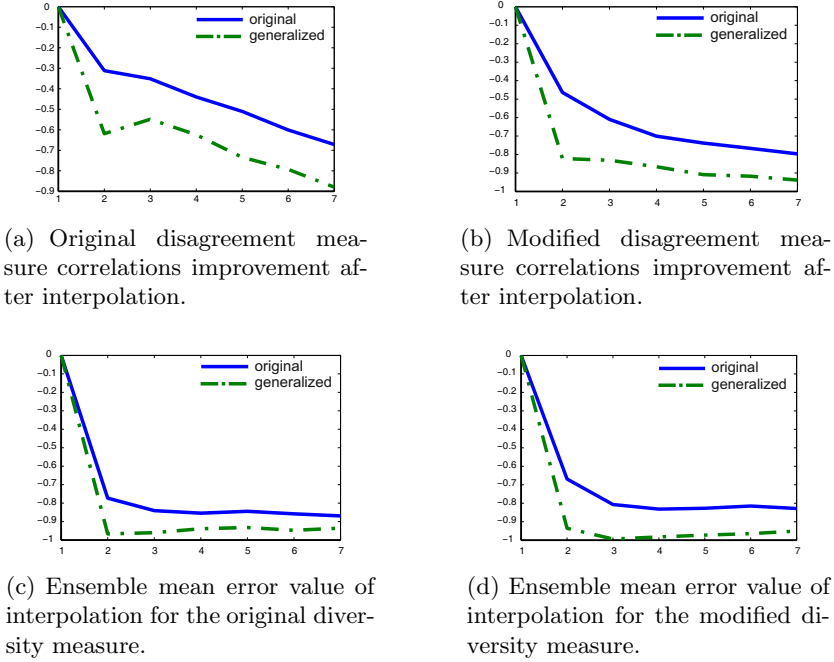


Fig. 4. Comparison of the diversity measures before and after interpolating the missing cases. After interpolation, the absolute value of the correlations are strongly higher, which is better for effectiveness of the diversity measures, with the system accuracy. The x-axis represents the number of the classifiers in the ensemble, while the y-axis the correlation value.

a system from the classifiers with the best performance regarding the diversity measures. This problem can be formulated as:

$$GD_j = \sum_{i=1}^I \alpha_{ij} d_{ij}, \quad j = 1, \dots, \binom{M}{k}, \quad k = (1, \dots, M),$$

where $\alpha_{ij} \in \mathbb{R}_{\geq 0}$ are the weights, d_{ij} are the values of the diversity measures, and GD_j is the value describing how good the specified system is considered as the diversity measures. In this case, the system with the maximal GD_j value will be choose:

$$GD = \max_j(GD_j) = \sum_{i=1}^I \alpha_i d_i.$$

The appropriate selection of the weights α_i are well-known from the literature for independent feature selectors. Namely, the optimal weights can be determined from the individual accuracies of the feature selectors [8]. In this special case, the correlation values show the performance of the diversity measures as feature

selectors. If we consider independent feature selectors (D_1, D_2, \dots, D_I) with accuracies (p_1, p_2, \dots, p_I) , then GD can be maximized by assigning the following weights

$$\alpha_i = \ln \frac{p_i}{1 - p_i}, (i = 1, \dots, I).$$

In our application, the accuracy p_i is the average correlation of the i -th diversity measure with the system accuracy regarding all possible assembled systems having the same number of members.

As an example for a special case because of the size of the full table, the optimal weights for the first nine diversity measures are shown in Fig. 5.

	DivM1	DivM2	DivM3	DivM4	DivM5	DivM6	DivM7	DivM8	DivM9
PossComb	2,73	4,02	5,01	2,98	2,46	3,84	2,10	3,62	1,93

Fig. 5. The applied weights in optimal weighted linear combination for the OD detection problem where the weights α_i were calculated as mentioned above. Every column contains a weight for a diversity measure (DivM) regarding a special case (PossComb).

In Fig. 6. the recommended combinations of algorithms for the weighted linear combination of the diversity measures are shown. It can be observed that the combined diversity measure(GD) well correlates with the system accuracy(Q).

Q (%)	Recommended combination (after inverse distortion)								GD
97,74	1	2	5	7	0	0	0	0	85,68
97,65	1	2	3	4	5	7	8	0	86,35
97,83	1	2	4	5	6	7	8	0	86,35
97,74	1	2	5	6	7	8	0	0	89,88
98,00	1	2	4	5	7	8	0	0	89,88

Fig. 6. The recommended combinations of the algorithms (expressed by sequential numbers of the algorithms in the middle of the table) using weighted linear combination of inverse distorted diversity measures. The first column (Q) shows the system accuracy, while the last column (GD) is the weighted combination of the diversity measures.

The ensemble system of the OD detector algorithms having the best accuracies can be found by applying our proposed method, and the selection can be made by GD value. The proposed weighted linear combination of diversity measures is novel for our extended model in the spatial domain.

6 Conclusion

In this paper the diversity measures introduced in classical majority voting are generalized for our voting model in spatial domain. We tested the generalized diversity measures for OD detection on the Messidor database of retinal images.

Without having any information about the dependencies among the applied algorithms, the aim is to choose the best ensemble system having the highest accuracy. In case of missing training data, interpolation should be applied. Moreover, the generalized diversity measures outperform the classic ones, and the most accurate ensemble system can be found by an optimally weighted combination of diversity measures. We tested our proposed method on the Messidor database [3] because it is the largest public dataset, the others contain not enough images to evaluate these measures properly.

Acknowledgments. This research was realized in the frames of TÁMOP 4.2.4. A/2-11-1-2012-0001 "National Excellence Program Elaborating and operating an inland student and researcher personal support system". The project was subsidized by the European Union and co-financed by the European Social Fund. This work was supported in part by the János Bolyai grant of the Hungarian Academy of Sciences; the project TÁMOP-4.2.2.C-11/1/KONV-2012-0001 supported by the European Union, co-financed by the European Social Fund; the OTKA grants K100339 and NK101680; the project TÁMOP 4.2.1./B-09/1/KONV-2010-0007 implemented through the New Hungary Development Plan, co-financed by the European Social Fund and the European Regional Development Fund.

References

1. Toman, H., Kovacs, L., Jonas, A., Hajdu, L., Hajdu, A.: A Generalization of Majority Voting Scheme for Medical Image Detectors. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part II. LNCS, vol. 6679, pp. 189–196. Springer, Heidelberg (2011)
2. Qureshi, R.J., Kovacs, L., Harangi, B., Nagy, B., Peto, T., Hajdu, A.: Combining Algorithms for Automatic Detection of Optic Disc and Macula in Fundus Images. *Computer Vision and Image Understanding* 116, 138–145 (2012)
3. Dataset MESSIDOR [Online], <http://messidor.crihan.fr>
4. Ruta, D., Gabrys, B.: Classifier Selection for Majority Voting. *Information Fusion* 6, 63–81 (2005)
5. Sharkey, A.J.C., Sharkey, N.E.: Combining Diverse Neural Nets. *The Knowledge Engineering Review* 12, 231–247 (1997)
6. Ruta, D., Gabrys, B.: Analysis of the Correlation Between Majority Voting Error and the Diversity Measures in Multiple Classifier Systems. In: *Proceedings of the 4th International Symposium on Soft Computing*, pp. 50–56 (2001)
7. Toman, H., Kovacs, L., Jonas, A., Hajdu, L., Hajdu, A.: Generalized Weighted Majority Voting with an Application to Algorithms Having Spatial Output. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012, Part II. LNCS, vol. 7209, pp. 56–67. Springer, Heidelberg (2012)
8. Kuncheva, L.I.: *Combining Pattern Classifiers, Methods and Algorithms*. John Wiley & Sons, Inc., New Jersey (2004)

How Do You Help a Robot to Find a Place? A Supervised Learning Paradigm to Semantically Infer about Places

Ioannis Kostavelis, Angelos Amanatiadis, and Antonios Gasteratos

Robotics and Automation Lab.,
Production and Management Engineering Dept.,
Democritus University of Thrace,
Vas. Sofias 12, GR-671 00 Xanthi, Greece
{gkostave, agaster}@pme.duth.gr, aamanat@ee.duth.gr
<http://robotics.pme.duth.gr>

Abstract. In this paper a visual place recognition algorithm suitable for semantic inference is presented. It combines place and object classification attributes suitable for the recognition of congested and cluttered scenes. The place learning task is undertaken by a method capable of abstracting appearance information from the places to be memorized. The detected visual features are treated as a bag of words and quantized by a clustering algorithm to form a visual vocabulary of the explored places. Each query image is represented by a consistency histogram spread over the memorized vocabulary. Simultaneously, an object recognition approach based on Hierarchical Temporal Memory network, updates the robot's belief of its current position exploiting the features of scattered objects within the scene. The input images which are introduced to the network undergo a saliency computation step and are subsequently thresholded based on an entropy metric for detecting multiple objects. The place and object decisions are fused by voting to infer the semantic attributes of a particular place. The efficiency of the proposed framework has been experimentally evaluated on a real dataset and proved capable of accurately recognizing multiple dissimilar places.

Keywords: place recognition, HTM, saliency map, semantics, robot navigation.

1 Introduction

Robotic semantic interpretation is an active research field, which exploits computer vision methods for place recognition [1]. A typical approach to simplify the information about a place is by treating the problem as a *bag-of-words* (BoW). These methods describe the input space as a collection of local features, the utilization of which, comprise a powerful representation for recognition [2]. However, since these features are randomly scattered in a scene, they should be stored in an ordered fashion so as to produce appearance based descriptions [3]. Some important works that utilize the BoW problem in robot navigation, are presented

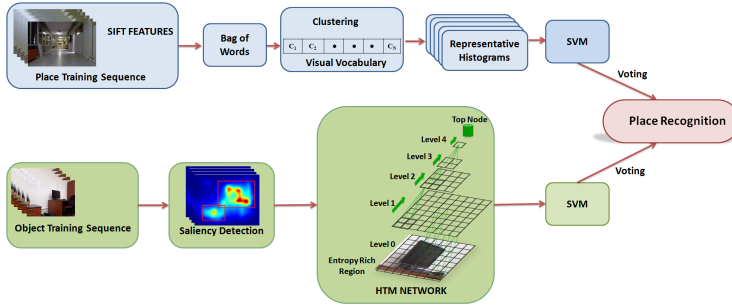


Fig. 1. The proposed place recognition algorithm

in [4], [5]. The authors in [6] proved that effective methods for object or place recognition are formed based on histogram-like descriptors.

Howbeit, during the robot’s locomotion in indoor environments, the utilization of solely such techniques for the place recognition frequently fails to come up with a reliable solution, due to the fact that the indoor scenes are usually cluttered and congested with numerous objects and unordered patterns. Thereupon, the need for acquiring additional information from visited places, which may derive from the objects in presence, is of great importance. Towards, this direction, the work in [7] utilize significant semantic characteristics for object recognition tasks, in order to form concept oriented representations of space, as well as to infer about the explored environment. The main disadvantage of such methods is that they assume simplifications which do not exhibit remarkable performance in cluttered scenes. Therefore, the interest has been turned into the bio-inspired systems, which try to imitate the human capabilities for recognizing a great variety of objects with little effort. Such a breakthrough came from the HTM theory described in [8], where the authors denoted that machine learning techniques should follow a hierarchical structure, similar to that of the brain. The HTM networks have already been exploited in numerous applications comprising supervised learning techniques [9], [10]. The work described in [11] employed the saliency detection as a preprocessing step, at the bottom level of the hierarchical network in order to comply with the human vision system.

Based on our previous work in this paper we propose a supervised learning method for recognizing places and objects during robot exploration in indoor environments, using multiple visual cues. It relies on appearance based histograms for the place recognition task, which are derived from solving a BoW problem employing a clustering method. A Support Vector Machine (SVM) classifier is trained to competently distinguish multiple classes. The object detection and recognition step is built on the saliency detection and HTM learning, constituting an integrated framework of our previous works described in [11] and [9], respectively. An SVM classifier is trained to learn the representative objects that correspond to the previously memorized places. The final decision about the currently visited place is taken by combining the output of the SVM classifiers for the place and object recognition in a voting fashion, exploiting the time

proximity of the acquired frames, while the robot explores an indoor environment. The steps of the proposed method are summarized in Fig. 1.

2 Place Recognition

The first subordinate module of the proposed work comprises a place recognition algorithm that produces preliminary inferences for the current visited places. One of the mandatory attributes that a robot should possess, is to effectively produce semantic inferences irrespectively to its current location. Due to its limited resources, the robot is able to memorize and recall only a finite number of representations about the explored space. In this work, we propose a spatial abstraction of the input space for the efficient memorization of the distinct places (e.g. “office”, “corridor”, etc.). In particular, the SIFT detection and description algorithm computes prominent points of a scene based on the appearance of the objects at particular interest points. Let us assume that the robot should learn different places from a queue of M images that contain various representations of such places. The SIFT algorithm is applied on every single image of the queue and the detected feature points are concatenated. The resulting feature space \mathbf{F} , turns out to a BoW that comprises a substantial description of the places that should be memorized.

Following the work presented in [12], \mathbf{F} is clustered by a vector quantization algorithm, namely the Neural Gas (NG) one. The latter is an artificial neural network and its basic objective is to optimize a cost function which minimizes the quantization error. In this work, NG has been preferred instead of the k-means to avoid local minimum solutions [3]. Thereupon, the set of Q centers of the resulting space quantization $\mathbf{C}^{128 \times Q} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_Q]$ comprises the visual vocabulary and provides a satisfactory representation of the initial space. The visual words are then utilized to create an appearance based histogram for each respective image of the sequence. Given the detected features, we form a representative consistency histogram $h_{\mathbf{S}_k} \in \mathbb{R}^Q$ for each image $k = 1, 2, \dots, M$ spread over the Q visual words. The L2 norm between the detected features and the visual words is calculated and the representative histogram is formed where the binning is performed according to the smaller distance. Consequently, each image in the sequence has been replaced by a respective appearance based histogram, which is utilized to execute further comparisons. The aforementioned procedure results in sparse histograms that vary significantly among the different classes and, therefore, its separation could be performed by a simple classification framework. Figure 2 depicts an example of this statement, where Fig. 2(a) and 2(b) depict an instance of the class “elevator area” and its respective histogram, while Fig. 2(c) and 2(d) presents an instance of the class “office” and its resulted histogram, respectively. It is clear that the formed histograms are significantly different, indicating the existence of variant patterns in the two distinct places.

The learning for the place category discrimination is accomplished by SVMs [13], which provide excessive performance of the SVMs in several visual recognition tasks [14]. Given the fact that the robot should learn various place categories, the one-against-all strategy has been preferred, i.e. for each different class

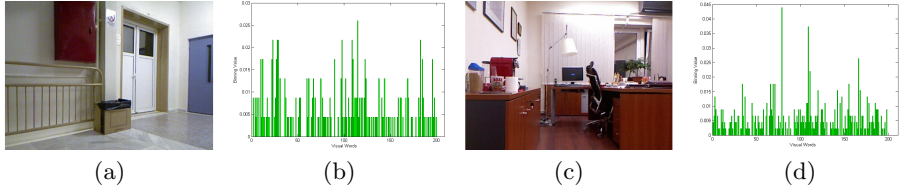


Fig. 2. a) Reference image indicating the class “elevator area”, b) the respective appearance based histogram, c) reference image indicating the class “office”, c) the respective appearance based histogram

a respective SVM is trained to separate it from all the others. The linear kernel was selected, due to the fact that it yielded remarkable recognition accuracy, while it kept low the number of parameters that have to be tuned.

3 Object Recognition

In order to perform object recognition tasks within a cluttered scene, a pre-processing step that allows the detection of the object and its separation from the other information of the image, is required. An almost straightforward procedure is the adoption of a saliency detection method capable of revealing the most prominent areas of an image, constituting a competent attentional model. Once the salient regions are detected within the scene, they are separated using a metric based on the entropy which operates directly on the saliency maps. The isolated region of images are introduced to the HTM network producing additional inference about the type of the detected object.

3.1 Object Detection Using Saliency Maps

The Graph-Based Visual Saliency (*GBVS*) algorithm was adopted for object detection [15]. In that method it is assumed that an image \mathbf{I} constitutes the superposition of the foreground \mathbf{f} and the background \mathbf{b} . The contribution of the foreground signal \mathbf{f} can be determined by taking the sign of the mixture signal \mathbf{I} in the transformed domain and, then, inversely transform it back into the spatial domain. The latter is achieved by computing the reconstructed image $\bar{\mathbf{I}} = IDCT[sign(I)]$, where *IDCT* is the *Inverse Discrete Cosine Transform*. Additionally, the foreground of an image, contrary to its background, is visually prominent. Therefore, the saliency map can be formed by applying a Gaussian kernel g to the reconstructed image $\bar{\mathbf{I}}$. The Gaussian filtering step is necessary since the salient objects are not only arbitrarily located in a scene, but they might also appear in a continuous region. The *GBVS* is decomposed into two consecutive steps. The first one is the *activation map* and comprises the formation of a fully connected weighted graph G by exploiting a simple dissimilarity measure over local neighborhoods of the images. In the next step, the activation

map is normalized and the weights between the nodes of G are computed. The output of GBVS algorithm is presented in Fig. 3, where the object detection algorithm is performed on a robot acquired image. The depicted image corresponds to a place with the “office” class label. The entropy based thresholding over the saliency map, reveals three connected components: a “coffee-machine”, a “desktop-screen” and the cluttered background. Both of the valid regions will be utilized to query the HTM, which will produce a semantic inference about their class type, whilst the cluttered background is disregarded.

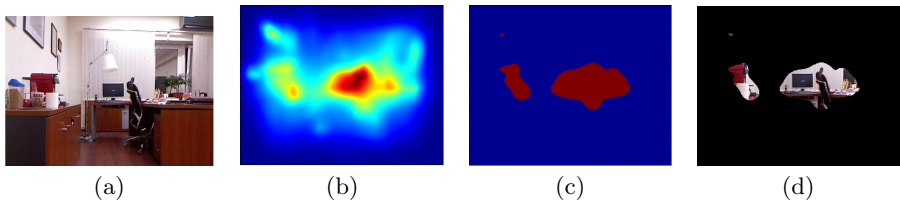


Fig. 3. a) The robot acquired reference image, b) the respective GBVS salient map superimposed on the color image, c) the resulted connected components after the entropy based thresholding, d) the respective regions of interest on the color image that will be fed as query images in the HTM network.

3.2 Hierarchical Temporal Memory Network

The output of the proposed object detection algorithm is subsequently utilized for the training of an HTM network. The adopted architecture closely resembles the one described in [11]. Each level of the network consists of adjoint nodes, the number of which decreases as the hierarchy increases. Formally speaking for the level ν the number of nodes is $2^{2\lambda-\nu}$, where λ denotes the number of the levels in the network. The *Level 0* is the input, i.e. the connected components (image portions that correspond to objects), which are presented to the network. These are divided into patches of n by n pixels. The nodes receive inputs from spatially-specific areas, namely the receptive fields and, therefore, they follow the same algorithmic procedures independent of the level they belong. Every single computational node undergoes two specific operation mechanisms. The first one constitutes, the training mode of the spatial module and the formation of the correlation one, whilst the second step subtends the inference mechanism, where the node produces outputs to be fed into the higher nodes.

Spatial Module. The input to the nodes of *Level 0* is the region inside a bounding box that contains the detected objects, as it has resulted from the aforementioned object detection algorithm. The spatial module has to learn a representative subset of the aforementioned image regions, which are expressed as input vectors in the receptive field of the network. The stored input vectors are the centers that encode the knowledge of the network. It is imperative that these centers should be carefully selected to ensure that the spatial module will

be able to learn a finite space of quantization centers from an infinite number of input vectors.

In the first step of the learning procedure, the initial input vector is considered as a quantization center at the respective node. Assuming that the learned quantization space in the spatial module of a node is $\mathbf{L} = [\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_N]^T$, where \mathbf{l}_i corresponds to quantization centers and N is the number of the existing centers, all the Euclidean distances d between these centers are calculated and their sum S is then computed:

$$S = \frac{1}{2} \sum_i^N \sum_j^N d(\mathbf{l}_i, \mathbf{l}_j) \quad (1)$$

For every new input vector \mathbf{l}_c presented in the receptive field of the node, all the distances d within the existing centers \mathbf{L} and the new input vector \mathbf{l}_c are computed. The sum S_c is then calculated:

$$S_c = S + \sum_i^N d(\mathbf{l}_i, \mathbf{l}_c) \quad (2)$$

The value of S represents the scatter of the existing quantization centers in the node. Any new input vector should be added in the node only when the within scatter S_c is much greater than the previous one. This approximation ensures that input vectors containing substantial information will be pooled in the spatial module, whereas those that do not contain representative information will be discarded. Therefore, for each new input vector the alteration ($alt = (S - S_c)/S$) between S and S_c should be examined against a threshold T . If $alt > T$ then, the query input vector becomes a new quantization center; otherwise, the next input vector is examined.

Correlation Module. The adopted measure of correlation is Pearson's coefficient, which obtain values in $[-1, 1]$. The correlation matrix is an N by N matrix containing the Pearson correlation coefficients between all the possible pairs of quantization centers. The \mathbf{R} is a symmetrical matrix as any quantization center is fully correlated with itself. The resulted correlation matrix is thresholded and only correlation values greater than 0.8 are kept. The next step of the proposed HTM is the partitioning of the correlation matrix into coherent subgroups. Each subgroup includes those quantization centers that share great coherence and, therefore, the resulted subgroups are utilized in the inference mode. Eventually, the input vectors of the nodes that lie in the upper level of the hierarchy are formed.

At this point it should be mentioned that the receptive field of *Level 1* of the network is 32 by 32 pixels. However, this does not constitute a functional restriction for the designed HTM network due to the fact that the detected objects of the images are fed solely to the retina of the network. Therefore, initial dimensions of the images are modified by the isolation of their connected component parts resulted from the salient regions of an image. The image portions

are then resized down to 32 by 32 pixels without losing significant amount of information. In the top node of the hierarchy a linear SVM is utilized capable of distinguishing the different type of classes that the objects may belong to. In addition, the HTM network might be instantiated more than once for each image based on the number of the detected objects in the current scene.

4 Experimental Results

This section deals with the evaluation of the accuracy of the proposed method. Each place that should be memorized is related with ordinary objects found in typical indoor environments. For example the “office” typically contains a “chair”, a “screen”, a “sofa”, while the place “corridor” contains “windows”, “doors”, “fire extinguishers”, etc. For the evaluation of the proposed dataset specific objects that appear to particular indoor places have been selected. Given that the SVM models are already trained off-line on a set of images, the place and object recognition modules are updated for each frame during the robot’s route. Moreover, the recognition of each place is fused among the outputs of the two different modules considering specific rules. The place recognition algorithm operates in a time window of $w = 20$ frames, within which the HTM inferences about the detected objects. The number of the recognized objects that belong to a specific object class within the time window is then calculated. Objects with high scores are selected for detailed examination in terms of the place that they belong. Consequently, the decision about the visited place is drawn according to the majority vote. In a similar fashion, the place recognition algorithm operates in the same time window and the decision about the visited place is governed by winner-takes-all rule. In cases that there is a draw, which means that the current frame is partially occluded or that the robot stands between two rooms, this frame is discarded. The final decision about the current place that the robot stands is taken by considering both the place and object recognition models. In case that there is a unanimous decision, the system examines the next frame. If the results between the two models diverges, the intersection between the most frequently appeared objects (in the time window w) and the second winner in the majority vote (in the object recognition algorithm) is also examined. If this scene is the one that the place recognition model firstly decided, then the system relies to the decision of the place recognition model, while in different case the current frame is discarded. The proposed method has been evaluated on a robot acquired dataset, utilizing the color camera of a Microsoft Kinect sensor [16]. It consists of three different parts; Part A includes samples shot under natural lighting conditions, Part B comprises samples shot under artificial lighting conditions and Part C constitutes a certain route that the robot has traveled in artificial illumination conditions. The place and object recognition algorithms have been both trained with Part A and B of the dataset and have been tested on the Part C, taking advantage of the time proximity during the decision procedure. The performance of the proposed algorithm is summarized in Fig. 4. In particular, Fig. 4(a) depicts the confusion matrix for the sole evaluation of

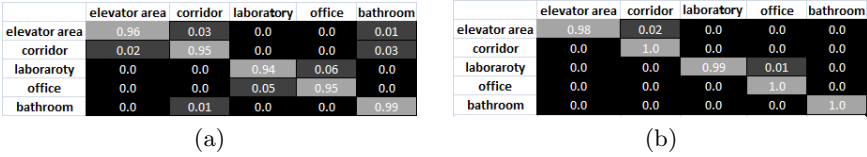


Fig. 4. a) The confusion matrix during the evaluation of the sole place recognition algorithm on Part C of the dataset, b) the confusion matrix during the evaluation of both the place and object recognition algorithm on the Part C of the dataset. Note that in the second case the classification accuracy has been improved significantly. The correct decisions are marked with green color while with red color are marked the erroneous ones.

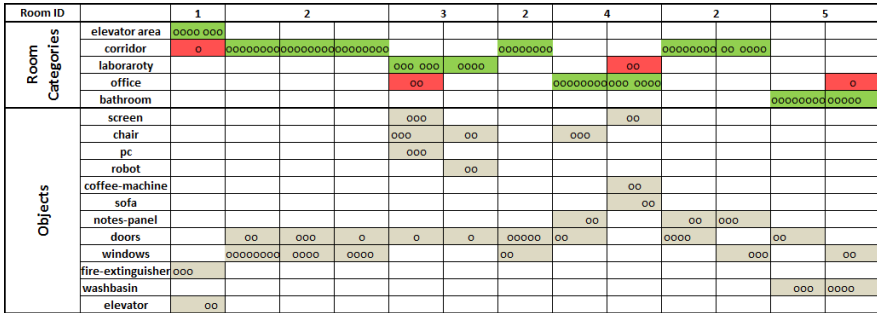


Fig. 5. Summary of the performance by visualizing the events registered by the system during exploration and the respective beliefs about the categories of the rooms, as well as the object’s presence.

the place recognition algorithm on the Part C of the dataset, while Fig. 4(b) depicts the confuse matrix also using the fused decision of the object recognition algorithm. Note that in the second scenario the performance of the system is superior, indicating that the additional information of the objects that belong in specific places increase the ability of the system to draw accurate semantic inferences. The experimental procedure is summarized in Fig. 5, where the performance of the proposed algorithm during the evaluation on the sequence Part C is exhibited. It visualizes the events registered by the system during exploration and its beliefs about the categories of the rooms as well the type of the detected objects. Each row represents the development of instantaneous decisions about a certain concept as the robot explored the environment, taking into consideration the specific time window and the result of the fusion procedure.

5 Conclusions

This work presented a robust place recognition algorithm, which integrates object recognition capacities for obtaining accurate semantic inferences during

robot exploration. The place recognition algorithm is based on spatial abstraction of the workspace, utilizing appearance based representative histograms. The object recognition task is decomposed into the detection and categorization subordinate modules. The detection routine is undertaken by the GBVS saliency map, which is accompanied with an entropy based thresholding metric that enables reliable detection of multiple objects within a scene. The object learning module is treated by a HTM network which entails great generalization capabilities. The semantic inference of visited areas is obtained by fusing the output of the place and object categorization modules in a hybrid manner exploiting the time proximity of the acquired frames. It should be mentioned that although the advantage of exploiting object recognition to enhance place recognition, was exhibited in this paper, the other way round, i.e. how the place recognition can help the object recognition is part of our future research. The proposed method has been evaluated on a robot acquired dataset and exhibited great performance of more than 98% classification accuracy in all cases. Consequently, the proposed method proved to be sufficient to draw accurate semantic inferences improving substantially the robot's navigation capabilities.

References

1. Pronobis, A., Martínez Mozos, O., Caputo, B., Jensfelt, P.: Multi-modal semantic place classification. *The International Journal of Robotics Research*, IJRR 29(2-3), 298–320 (2010)
2. Torralba, A., Murphy, K., Freeman, W., Rubin, M.: Context-based vision system for place and object recognition. In: *International Conference on Computer Vision, ICCV 2003*, pp. 273–280. IEEE (2003)
3. Zhang, J., Marszałek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision* 73(2), 213–238 (2007)
4. Filliat, D.: A visual bag of words method for interactive qualitative localization and mapping. In: *International Conference on Robotics and Automation, ICRA 2007*, pp. 3921–3926. IEEE (2007)
5. Fraundorfer, F., Engels, C., Nistér, D.: Topological mapping, localization and navigation using image collections. In: *International Conference on Intelligent Robots and Systems, IROS 2007*, pp. 3872–3877. IEEE (2007)
6. Fazl-Ersi, E., Tsotsos, J.: Histogram of oriented uniform patterns for robust place recognition and categorization. *The International Journal of Robotics Research* 31(4), 468–483 (2012)
7. Vasudevan, S., Siegwart, R.: Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robotics and Autonomous Systems* 56(6), 522–537 (2008)
8. Hawkins, J., Blakeslee, S.: *On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines*. Henry Holt & Company, New York (2004)
9. Kostavelis, I., Gasteratos, A.: On the optimization of hierarchical temporal memory. *Pattern Recognition Letters* 33(5), 670–676 (2012)
10. Charalampous, K., Kostavelis, I., Amanatiadis, A., Gasteratos, A.: Sparse deep-learning algorithm for recognition and categorisation. *Electronics Letters* 48(20), 1265–1266 (2012)

11. Kostavelis, I., Nalpantidis, L., Gasteratos, A.: Object recognition using saliency maps and htm learning. In: IEEE International Conference on Imaging Systems and Techniques, IST 2012, pp. 528–532. IEEE (2012)
12. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV 2004, vol. 1, p. 22 (2004)
13. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
14. Pronobis, A., Caputo, B., Jensfelt, P., Christensen, H.I.: A realistic benchmark for visual indoor place recognition. Robotics and Autonomous Systems 58(1), 81–96 (2010)
15. Hou, X., Harel, J., Koch, C.: Image signature: Highlighting sparse salient regions. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(1), 194 (2012)
16. Kostavelis, I., Gasteratos, A.: Cognitive Navigation Dataset, Group of Robotics and Cognitive Systems (2012), <http://robotics.pme.duth.gr/kostavelis/Dataset.html>

Study of the Pre-processing Impact in a Facial Recognition System

Guillermo Calvo¹, Bruno Baruque¹, and Emilio Corchado²

¹ Department of Civil Engineering, University of Burgos, Spain
gco0000@alu.ubu.es, bbaruque@ubu.es

² Department of Computer Sciences and Automatic, University of Salamanca, Spain
escorchado@usal.es

Abstract. The present work is a study of the influence of the preprocessing stage on the classification performance of a face recognition analysis. To carry out this task have made tests in a full FRS, evaluating each of its four stages and including several advanced alternatives in preprocessing, such as illumination normalization through the *Discrete Cosine Transformation* or alignment by *Enhanced Correlation Coefficient*, among others. The main goal of this work is determining how those different preprocessing alternatives interact with each other and in which degree they affect the overall Facial Recognition Systems (FRS). The tests make a special emphasis in using images that could have been obtained from a real environment, rather than at a lab environment, with the difficulties that this brings for facial recognition techniques.

Keywords: Face Recognition, Preprocessing, Normalization, Alignment, ECC, DCT.

1 Introduction

A classification system is greatly influenced by data preprocessing and much of its success lies in selecting the best techniques for the task performed. As face recognition can be essentially put on the same level as a regular classification problem, it faces a similar challenge in this first stage. However, due to the nature of the images and the people involved, the data will inevitably include higher variations (or noise) than a classic classification problem. The images to analyse are influenced by other aspects intrinsic to the human physiognomy such as those due to the attitude of the model represented, changes in stands, gestures, clothing, hats, distance, tattoos, prosthetics or changes in appearance.

In addition, there are technical factors that increase the complexity of the classification. Many of them are due to image capture systems, associated systems lighting (flash) or kind of data from where images are obtained (still image, video, 3D, infrared, etc).

Finally, there are also external factors related to the environment such as the light conditions, background image, temperature or presence of other people, among others.

It is not always possible to work in a controlled environment so that images are not affected by the factors described above. However, not testing the system under these natural conditions will yield unreliable results.

There are several publications that conduct surveys on this kind of systems, but either they are centred in summarizing the results of the original publications [1] or they refer to specific problems observed in experimental tests [2,3]. Although these kind of studies are very interesting, it is also very informative to test the importance of the influence of all phases on the complete process of the final recognition of individuals. According to the knowledge of the authors, these kind of studies have not been carried out very often in a practical manner. In this case, the study is especially focused on the preprocessing of images, being this a crucial stage in the process.

2 Face Recognition Architecture

A Facial Recognition System is composed of five stages as it was discussed in [4]. These include the following ones (see also Fig. 1):

1. Image Capture
2. Facial Detection
 - Finding faces
 - Selection and image adjustment
3. Preprocessing
 - Illumination Normalization
 - Image Alignment
4. Feature Extraction¹
5. Recognition process (classification)

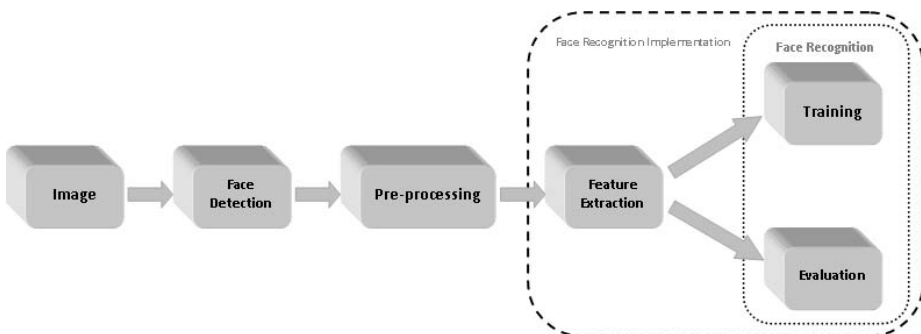


Fig. 1. Main scheme of a Face Recognition System

The capture of images rises the problems discussed in the introduction and a facial recognition process must be prepared to face them. The next step is to

¹ It is usually performed with the recognition process stage.

discriminate the number of faces detected in the image. During this process the image is cropped to the size of the selected face and its size is standardized.

The resulting images must be preprocessed in order to obtain data to be easily classified. To do so, the light is homogenized to prevent glare or shadows areas that transform the image appearance. Another process within preprocessing is the alignment, which aim is assuring that the location of the face in a picture is the same in all pictures belonging the same class, in order to align facial features in all the images of the same person.

The stages of feature extraction and recognition are strongly interrelated and are usually included in one single step in practical approaches.

3 Techniques Used

3.1 Facial Detection

The *locally SMQT features & split up SNoW classifier* [5] has been used as the face detection technique in this study. This technique is based on appearance and is divided into two steps, the first is the use of SMQT [15] (Successive Mean Quantization Transform) locally in order to obtain the illumination invariant features and then, as second step, to use a SNoW (Sparse Network of Winnows) classifier.

- **SMQT**[15]. This technique improves the image quality making it insensitive to the gain and the bias (off-set), considering an image that can be influenced by those factors (see eq. 1).

$$I(x) = gE(x)R(x) + b \quad (1)$$

The reflectance (R) has the the structure of the image itself, and it is needed to make the image invariable to gain (g) and bias (b) in order to be able to extract R and considering E as a *constant*.

- **SNoW Classifier**[5]. This classifier get the features obtained by the previous step and uses a network of linear units to define the space of learned characteristics. To do so, it uses an initial SNoW classifier and the results from it are subdivided into other SNoW classifiers.

In the tests performed, this enables to crop only the face area detected, avoiding head shapes, hair, ears...etc.

3.2 Preprocessing

The purpose of this stage is to eliminate those features that hinder the classification of images of the same person (intra-class differences), thereby increasing the difference of them with others (inter-class differences).



Fig. 2. Image preprocessing secuency (*Initial*→ *Face Recognition*→ *DCT*→ *ECC*)

Illumination Normalization. *Discrete cosine transform in logarithm domain* [6] along with the work of [7] [8] and a normalization method *based on RGB* [9] have been chosen as illumination normalization methods. The first one is based on a discretization of cosine as opposed to the second, which is based on a histogram with much milder changes in the values of illumination. Both methods perform a normalization of the full face image.

- **Discrete Cosine Transform (DCT)**[6]. This normalization technique is based on the discrete Fourier transform, but using only real numbers. The procedure of this method is to adjust the dynamic range of a image in grayscale to the interval [0-255] and then truncate the ends of the image histogram². This operation allows distributed gray levels along a image, eliminating the problems of existence of very bright values in the image that could dark the rest of the image after size changing. Finally, a photometric image normalization is done. The technique establish a predefined number of DCT coefficients to zero, eliminating the low frequency part of the data. This low frequency information is believed to be susceptible to changes in illumination.
- **RGB pixel compensation**[9]. This method uses an adaptive illumination compensation, based on the black pixel, through a histogram equalization of the image. This is a two step method in which the first RGB image is compensated and then converted to YCbCr in order to normalize the image illumination.

Image Alignment. Regarding the alignment techniques, *Enhanced Correlation Coefficient Maximization* [10] and *eye detection alignment* [11], which makes an alignment through the eye positions; have been chosen for this study. The first one was selected because is based on a template in contrast with the second one.

² It removes a certain percentage of the low and high end of the histogram of an image.

- **Enhanced Correlation Coefficient (ECC)**[10]. The algorithm takes two images (an input image and an image template) as input, and estimates the 2D geometric transformation between them. It is possible to adjust several parameters in this algorithm, such as enabling implementation on a number levels on a pyramid scheme or without it, the number of iterations per level, choose the type of transformation or using an initial transformation matrix.
- **Alignment through eye coordinates**[11]. The first step is to detect the eyes on a face using the cascade of Haar features. After obtaining those coordinates a transformation of the image is made to align it through a spatial transformation taking two checkpoints from the image.

3.3 Facial Recognition

Among the options for the recognition algorithm selected to construct the FRS for conducting this study, three different techniques have been included. Two holistic methods: *Eigenfaces*[12], as one of the most widespread algorithms on this category, and *Fisherfaces*[13] which is an well-known evolution from the previous one. And, to broaden the scope of the study, a feature based method such as *Hidden Markov Model*[14] has also been selected.

- **Eigenfaces**[12] (Holistic). It is a classification method based on Principal Component Analysis (PCA) to reduce the dimensionality³ of each image and projected their attributes on the new dimensions considered. Finally, the final classification is obtained by comparing the Euclidean distances between the data obtained for each image. This technique provides a reasonably satisfactory results [12] and has low computational load.
- **Fisherfaces**[13] (Holistic). The Fisherfaces method is based on LDA (Linear Discriminant Analysis) and it uses information between members of the same class to develop a set of feature vectors where the variations between the different faces (or classes) are emphasized while the differences within the same class are minimized. Previously, Fisherfaces uses PCA to reduce the dimension of the data, as can be seen on Eq. 2, where S_B is the scattering matrix between classes, S_T is the intra-class ones and W is the orthonormal matrix of the new space. The results are better than those achieved just with PCA, without preprocessing as varying lighting conditions or with slight changes in facial expressions [13].

$$W_{opt} = arg \max \frac{|W^T S_B W|}{|W^T S_W W|} \quad (2)$$

- **Hidden Markov Model (HMM)**[14](Feature Based). A Hidden Markov Model, is a statistical model which assumes that the system model is a Markov process of unknown parameters and could be considered as a dynamic Bayesian simple network. This algorithm is based on the division of

³ Having selected 50 training images the number of principal components used in this algorithm is 49.

7 facial image regions (7 nodes) and a state transition probability. The algorithm provides a probability that a given region of the face will follow another with certain determined features. According to that probability, the analysed image is associated or not to the considered matching image. This algorithm previously uses a histogram equalization (HE).

4 Experiments and Results

To establish a valid test methodology, given that the problem to solve is a multiclass classification one, a comparison of One-Vs-All (OVA) is performed, as suggested in [16]. In all tests performed, a cross-validation 6 K-fold [17] for 10 different people (or classes) has been performed. The clusters are composed by a set of 60 images (6 per person), using 50 for training and 10 for testing. The confidence interval values are represented by percentage of success $(1 - E) \cdot 100$ and using the mean error obtained through $E = \frac{1}{K} \sum_{i=1}^k E_i$, where E is the error percentage and k the number of folds in the cross-validation.

4.1 Databases

Two different image databases have been used for testing. The first is the **Caltech** [18] one which shows people with different image backgrounds, light conditions, facial expressions and camera distances. These images are not pre-processed, maintaining consistency with those we could take in any environment, which increases the difficulty for a FRS discriminating among them ⁴. The second database used on this work is **ORL** [19], where pictures have all the same image background, are focussed at heads and, although they are taken at the same distance, the pose variation is much bigger than those used in *Caltech*.

4.2 Experiments

Two experiments have been performed for this study. The first one has been composed as the most complete test possible, using all combinations available to construct the FRS. Results obtained in this experiment have been confirmed in the second one, with the use of a different database and choosing the combinations with the best results obtained in the first experiment.

Procedure Experiment 1. The images have been transformed to grayscale when it was necessary, also they have been resized to 179x118 pixels in its initial stage and to 46x46 pixels from the face detection process in order to reduce the computational load.

The objective of this experiment is to observe how the recognizing accuracy of the system increases or decreases for each of the classifiers selected (see 3.3) when additional stages are added to the FRS. In order to observe this variations, the following test configurations have been included in the test:

⁴ The high intra-class variation increases the difficulty of classification methods that rely on common features in each class (Fisherfaces).

- Original images only (*INITIAL*)
- Cropped facial images -only face- (*FACE DETECT.*)
- DCT illumination normalization algorithm (*DCT*)
- RGB illumination normalization algorithm (*NORM.RGB*)
- ECC alignment system (*ECC*)
- Eye Position alignment system (*EYE.ALIGN*)
- DCT illumination correction + ECC alignment (*DCT + ECC*)
- DCT illumination correction + Eye-align (*DCT + EYE.ALIGN*)
- Norm-RGB illumination correction + ECC alignment (*NORM.RGB + ECC*)
- Norm-RGB illumination correction + Eye-align (*NORM.RGB + EYE.ALIGN*)

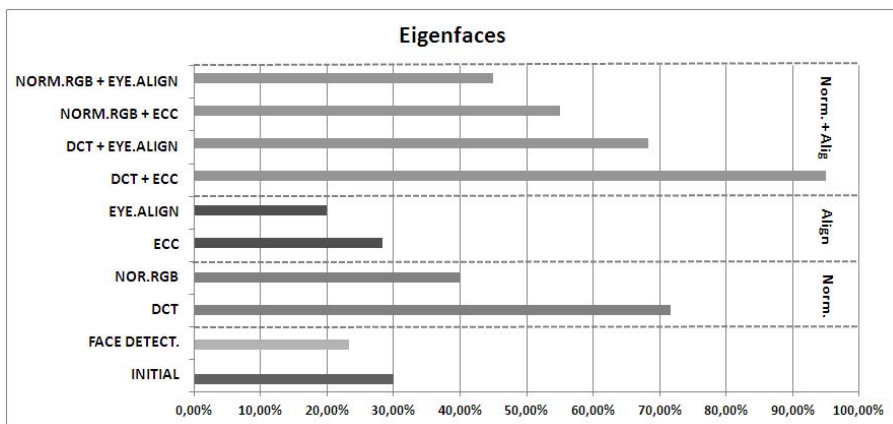


Fig. 3. Eigenfaces Results

Results Experiment 1. The results obtained in this test are shown in Figs. 3, 4 and 5. In them, the corresponding confidence rates are shown.

Comparing results from the initial stage to those including face detection ones show that:

- The results for eigenfaces (see fig. 3) are worse to others because in all processes the outline of the head, which should help this particular classifier to discriminate people, has been removed from the pictures.
- For fisherfaces (see fig. 4) the results are greatly improved (from 8.33% to 46.66%) because the gain of minimizing similar data inside each class due to same image backgrounds is bigger than the loss due to the absence of head shape.
- In the HMM model (see fig. 5) its results are also highly improved (from 18.33% to 63.33%), because this classification technique is based on features and the elimination of the image background increases their discrimination based on probabilities.

Comparing the results from the combinations using only the face detection step to those including illumination normalization, shows that:

- For eigenfaces (see fig. 3) there is a high increase in the success rate, which goes from 23,33% to 71,66% for the best technic selected (DCT), due to the improving of data (avoiding hidden areas) making them linearly divisible.
- In fisherfaces (see fig. 4) the rate of success decrease because some distinguishing features between classes are erased by illumination normalization process.
- For HMM (see fig. 5) there is a slight improvement in the success rate (increment 10%), though this happens just with the best method (DCT). This is due to improved of using both DCT and the pre-filter algorithm itself (see 3.3).

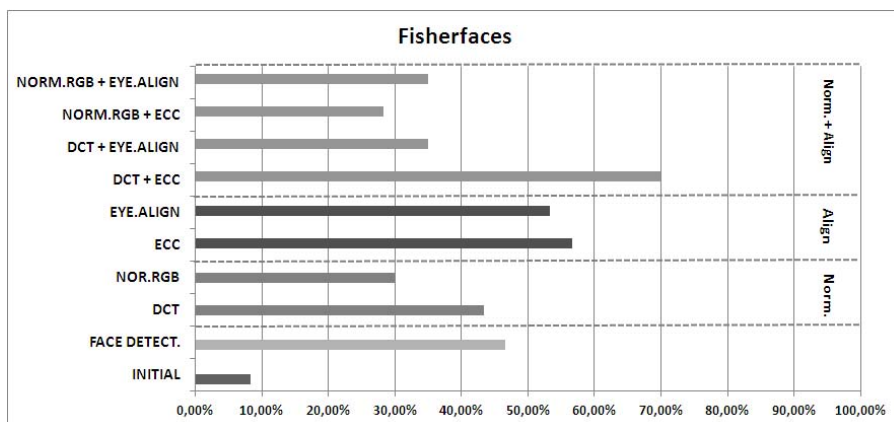


Fig. 4. Fisherfaces Results

In the results for combinations using the normalization step without a previous illumination normalization show that:

- For eigenfaces (see fig. 3) shows just very slight improvement rates or even some worse with the technique (EYE.ALIGN). That indicates without a previous illumination normalization the alignment does not increase the separation between classes.
- In fisherfaces (see fig. 4) the image alignment stage, however, no worse affects and even get better results in both techniques (ECC y EYE.ALIGN) going from 46,66% to 56,66%.
- HMM (see fig. 5) does not show improvement in its performance and even there is a slight decrease for the worst method (EYE.ALIGN) by the introduction of black pixels zones inside images in this step, this is something which makes difficult to HMM to discriminate between classes.

The last combination set, including a illumination normalization and then an alignment method, show that:

- Eigenfaces (see fig. 3) shows better results than any of the other combinations for (DCT + ECC). The alignment prior and subsequent illumination normalization makes first increase the separation between classes and then decrease the distance within each class.
- In fisherfaces (see fig. 4) the combination of illumination normalization and then applying alignment makes the results improve to 70% for (DCT + ECC).
- For HMM (see fig. 5) there are not significant differences in the values obtained and show a substantial decrease for the worst combination (NORM.RGB + EYE.ALIGN).

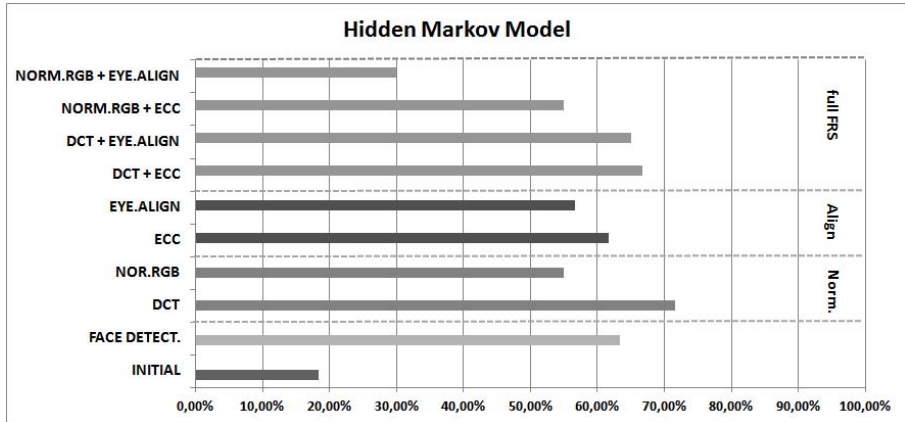


Fig. 5. Hidden Markov Model Results

It is obvious when analysing these results that the preprocessing stage greatly affects final results for eigenfaces classifiers and that the biggest impact within it is the illumination normalization.

It can also be observed that preprocessing decisively affects less fisherfaces than eigenfaces, since images are standardized and make fisherfaces lose its more defining quality: to increase inter-class differences (which are now much less evident), obtaining worse results when explicit different position variation is included within a single class (high intra-class variability in initial images).

It can also be noted that the results for the HMM method are not strongly influenced by preprocessing stages and that the algorithm is able to get acceptable results without the use of this stage, since it is based on features, and analyzing them separately is not severely affected by large areas of lighting changes or alignment.

Procedure Experiment 2. In this experiment, images are initially in grayscale and resized to 46x46 pixels from the face detection process.

For this test all mentioned algorithms from the stage of face detection to the illumination normalization and alignment have again been considered. This time, only the best combination of each stage has been used (*Face Recognition* → *DCT* → *ECC* → *DCT+ECC*). The purpose of this second study is to corroborate the results obtained in the *experiment 1* (see Section 4.2).

Table 1. Comparison for the best solution with ORL images

	Face-Detect.	Norm-DCT	Align-ECC	DCT+ECC
Eigenfaces	60%	70%	71%	93%
Fiserfaces	55%	31%	65%	60%
Hidden Markov Model	85%	60%	68%	67%

Results Experiment 2. As seen from the data given in Table 1, classification methods which are inherently better avoiding *noise* in lighting, position or gestures (fisherfaces and HMM) get worse results than eigenfaces after preprocessing. This emphasizes the importance of preprocessing methods used when the classifier is not oriented on differences between classes or features. These results are consistent with those in the previous experiment and reflect equally the importance of preprocessing for FRS.

5 Conclusions

It follows from the results obtained that some preprocessing methods do not work properly with the classifiers that take into account the knowledge of the number of classes (Holistic) or are not so critically dependent on it for their operation (Features Based). However, methods such as eigenfaces crucially improve their classification capabilities with a suitable preprocessing.

Feature based or holistic methods designed to avoid intrinsically variations introduced by lighting or pose, are most robust in results without a prior preprocessing, being these more suitable than methods that have not been developed with this purpose (such as eigenfaces).

With the existence of many variations in image conditions for the same class (which implies a great intra-class variability) may be more beneficial to use a preprocessed and holistic classifier than using a discriminatory or advanced classifier (such as fisherfaces or HMM), because the preprocessing makes the classes are linearly separable without leaving that task to the last stage of the FRS.

When the conditions of image capture do not include a controlled environment, classification methods find serious problems in getting good levels of confidence that maybe a good preprocessing could fix it.

The preprocessing stage is not just important, but also delicate, it is essential to know the functioning of the classifier used to choose an appropriate preprocessing to improve the results, in case of mistake its impact could be negative.

Acknowledgements. This research is partially supported by the Spanish Ministry of Economy and Competitiveness under project TIN2010-21272- C02-01 (funded by the European Regional Development Fund), SA405A12-2 from Junta de Castilla y León. This work was also supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

References

1. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. *ACM Computing Surveys* 35, 399–459 (2003)
2. Zhang, X., Gao, Y.: Face recognition across pose: A review. *Pattern Recognition* 42, 2876–2896 (2009)
3. Sang-Il, C., Chong-Ho, C., Nojun, K.: Face recognition based on 2D images under illumination and pose variations. *Pattern Recognition Letters* 32, 561–571 (2011)
4. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. Addison-Wesley Pub. (Sd) (2008)
5. Nilsson, M., Nordberg, J., Claesson, I.: *Face Detection using Local SMQT Features and Split up Snow Classifier* (2007)
6. Chen, W., Er, M.J., Wu, S.: Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. *IEEE Transactions on Systems, Man and Cybernetics, Part B* (2006)
7. Štruc, V., Pavešić, N.: Photometric normalization techniques for illumination invariance. In: *Advances in Face Image Analysis: Techniques and Technologies*. IGI-Global (2011)
8. Štruc, V., Pavešić, N.: Gabor-based kernel-partial-least-squares discrimination features for face recognition. *Informatica (Vilnius)* (2009)
9. Chen, L., Grecos, C.: Fast skin color detector for face extraction. *Electronic Imaging* (2005)
10. Evangelidis, G.D., Psarakis, E.Z.: Parametric Image Alignment Using Enhanced Correlation Coefficient Maximization. *IEEE Transactions on Systems, Pattern Analysis and Machine Intelligence* (2008)
11. Everingham, M., Sivic, J., Zisserman, A.: “Hello! My name is... Buffy” – Automatic Naming of Characters in TV Video. In: *Proceedings of the British Machine Vision Conference* (2006)
12. Turk, M., Pentland, A.: Eigenfaces for Recognition. *Journal of Cognitive Neuroscience* (1991)
13. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection (1997)
14. Miar-Naimi, H., Davari, P.: A New Fast and Efficient HMM-Based Face Recognition System Using a 7-State HMM Along With SVD Coefficients. *Iranian Journal of Electrical & Electronic Engineering* (2008)
15. Nilsson, M., Dahl, M., Claesson, I.: The successive mean quantization transform. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2005* (2005)
16. Rifkin, R., Klautau, A.: In Defense of One-Vs-All Classification. *Journal of Machine Learning Research* (2004)
17. Refaeilzadeh, P., Tang, L., Liu, H.: Cross-Validation. In: *Encyclopedia of Database Systems*. Springer US (2009)
18. Caltech Computational Vision Group. Faces 1999 Database, <http://www.vision.caltech.edu/html-files/archive.html> (last accessed: 2012)
19. AT&T Laboratories. Cambridge ORL Faces Database, <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html> (last accessed: 2012)

Using ABC Algorithm with Shrinkage Estimator to Identify Biomarkers of Ovarian Cancer from Mass Spectrometry Analysis

Syarifah Adilah Mohamed Yusoff^{1,2}, Rosni Abdullah¹, and Ibrahim Venkat¹

¹ School of Computer Sciences, Universiti Sains Malaysia
{rosni,ibrahim}@cs.usm.my

² Dept Computer Sciences and Mathematics,
Universiti Teknologi MARA Pulau Pinang, Malaysia
syarifah.adilah@ppinang.uitm.edu.my

Abstract. Biomarker discovery through mass spectrometry analysis has continuously intrigued researchers from various fields such as analytical researchers, computer scientists and mathematicians. The uniqueness of this study relies on the ability of the proteomic patterns to detect particular disease especially at the early stage. However, identification through high-throughput mass spectrometry analysis raises some challenges. Typically, it suffers from high dimensionality of data with tens of thousands attributes and high level of redundancy and noises. Hence this study will focus on two stages of mass spectrometry pipelines; firstly we propose shrinkage estimation of covariance to evaluate the discriminant characteristics among peaks of mass spectrometry data for feature extraction; secondly a sophisticated computational technique that mimic survival and natural processing which is called as Artificial Bee Colony (ABC) as feature selection is integrated with linear SVM classifier for this biomarker discovery analysis. The proposed method is tested with real-world ovarian cancer dataset to evaluate the discrimination power, accuracy, sensitivity and also specificity.

Keywords: metaheuristic, feature selection, swarm algorithm, bio-inspired algorithm, classification, feature extraction.

1 Introduction

The well-known soft-ionization techniques such as Matrix-Assisted Laser Desorption/ Ionization Time-of-flight Mass Spectrometry (MALDI-TOF-MS) and Surface-Enhanced Laser Desorption/Ionization Time-Of-Flight Mass Spectrometry (SELDI-TOF-MS) have created beautiful insights towards high-throughput proteomics analysis to the researchers across multi-disciplines. It works on the principle that different molecules have different masses, thus once a substance is injected to the instrument, the constituent can be separated according to their masses. Interestingly, this output patterns are able to exhibit structures of proteins, characterization of regulatory and functional networks, investigation

of molecular defect in biological fluids and identification of various stages of a disease via development of reagents [3]. The biological interest on this study focuses on biomarker identification through the expression of proteins which can diagnose and prognose markers for the disease.

Typical output data of mass spectrometry yields a spectrum which consists of mass to charge ratio (m/z) on x-axis and ionisation intensity on y-axis. Significant information of the spectrum comprises peaks of the intensities with proportion to m/z values. The underlying information pertaining to the peaks that represent particular proteins or peptides would lead to discovery of new biomarkers for particular disease on different stages [12]. However, mass spectrometry data that suffer from high dimensionality will degrade the classification performance due to few data samples in a high dimensional variable space. Therefore, this study will focus on two different stages of mass spectrometry pipelines: (1) Feature extraction - we will assemble and calibrate all detected peaks across different sample through shrinkage estimator. This stage will simultaneously reduce dimensionality as we choose only the relevant peaks; (2) Feature selection - we adapt foraging behaviour among bees for optimising and search only parsimonious features through a learning model.

According to [16], list of biomarkers is stabled when some features are strongly correlated to each other and equally relevant for the task at hand. Furthermore, overlapping features happen when high correlated features are possibly being selected differently in different setting [6]. In mass spectrometry analysis, both assembling and calibrating peaks are methods that grouped or coalesced some neighbour peaks together in order to extract a set of highly correlated and independent features. Armananzas et al. [1] have used linear correlation method to assemble peaks and group them under same peaks-bin. Meanwhile, Resson et al. [15] have proposed peak calibration from the idea of Coombes et al. [4] by combining peaks in the range of 7 clock ticks or at most 0.03 percent relative mass. However approach proposed by [15] is not appropriate to deal adequately with variety datasets across different soft-ionization platforms.

In many situation of statistical analysis, estimating the population covariance matrix is inevitable and typically estimated by the sample covariance matrix, S_{ij} .

$$S_{ij} = \frac{1}{n-1} \sum_{k=1}^n (X_{ki} - \bar{X}_i)(X_{kj} - \bar{X}_j) \quad (1)$$

Where X_{ki} and X_{kj} is the k-th observation of the variable X_i and X_j respectively. Meanwhile the mean of \bar{X}_i is defined as:

$$\bar{X}_i = \frac{1}{n} \sum_{k=1}^n X_{ki} \quad (2)$$

Both observations and variables in this study referring to number of samples and features of the data, respectively.

The beauty of covariance estimator is, it is the only estimator which well-conditioned and has accurate characteristic among the other statistical methods [11]. However, it would be challenging for a covariance matrix to interpret sample with that holds less number of samples than features ($n < p$). This situation

occurs especially in bioinformatic domains such as gene expression data and mass spectrometry data. The estimation of covariance matrix is not possible or not accurate due to violation of the two conditions required for estimation: positive definite and well-conditioned. All the eigenvalues of covariance matrix should be non-zero to be positive definite and they are well conditioned when invertible.

Ledoit and Wolf [10] have proposed several approaches through shrinkage estimator that estimate the covariance matrix. These innovative approaches has opened new paradigm that involve data with huge features and small sample sizes. These studies focus on finance profession problems and have efficiently been enhanced into life science in [17,19]. Schafer et al. [17] applied shrinkage estimator for covariance and correlation matrix to infer gene network, while Yao et al. [19] proposed shrinkage correlation coefficient as similarity matrix for clustering replicate gene microarray data. Both studies have performed not only much better than comparing methods but claimed to be positive definite and invertible. Implicitly, calculating covariance by shrinkage estimation of covariance is more suitable and statistically efficient to evaluate the discriminant characteristics among peaks of mass spectrometry data for feature extraction.

Recently, researchers have been allured with some biological or natural life style and imitate the process to be adapted in solving real life problems. With specific emphasis to optimisation problems, these biological inspired algorithms are mainly constructed based on the modern metaheuristic paradigm. It composes exploration and exploitation behaviour of certain living organism such as ant colony, foraging bees, fish schooling and immune systems. Artificial BeeColony (ABC) has been proposed by Karaboga [8] to solve numerical optimisation. This study has performed significantly compared to others population-based search algorithms. Constructed based on foraging bees concepts, this algorithm has been continuously applied to various optimization problems and become matured every year with subsequent improvements. Interestingly, Karaboga and Ozturk [9] have applied the ABC in data mining with specific focus to clustering several datasets from UCI database. Further, [14] have adapted the ABC algorithm for classification purposes by injecting new rules to suite with classification. At the same time, [2] have improved ABC algorithms by applying a new heuristic classification. Hence, this study investigates the suitability of feature extraction techniques to be coupled with ABC as feature selection in order to improve performance of classification and produce reliable results of biomarkers identifications.

This paper is organised as follows: Section 2 elaborates the fundamental of shrinkage approaches for feature extraction; Section 3 introduces ABC as feature selection; Section 4 discusses the implementation and results of high resolution ovarian dataset through the whole mass spectrometry pipelines, purposely for comparison of shrinkage covariance correlation estimation with empirical covariance correlation.

2 Shrinkage Estimator

This section presents the shrinkage estimator for covariance matrix proposed by James and Stein [7] to assemble and calibrate the peaks to be well discriminated upon features extraction. The idea of shrinkage implies that the sample of covariance is shrunk towards structured estimator. The equation for shrinkage estimator is presented as follow:

$$\mathbf{S} = \alpha T_{ij} + (1 - \alpha)S_{ij} \quad (3)$$

The main components in the shrinkage estimator are both sample covariance matrix denoted by S_{ij} and highly structured estimator that are denoted by T_{ij} . Compromise of both components is considered as convex linear combination of \mathbf{S} , where α is shrinkage constant between 0 and 1 and is defined as:

$$\alpha = \max\left(0, \min\left\{1, \frac{k}{p}\right\}\right) \quad (4)$$

Shrinkage estimator has varied techniques for the structured estimator (shrinkage target) and shrinkage constant [17]. Shrinkage target is measured as the prior information, where in this study shrinkage target which is based on single-index model covariance matrix has been applied as proposed by Ledoit and Wolf [10]. However, single index model is severely biased, though it composed only few estimator errors. Thus, they have used properly weighted average as optimal trade-off between bias and estimator error.

The major concern in this estimation is optimal selection of shrinkage intensity. Existing shrinkage estimators [5] are broken down when $n < p$ because of their loss function depends on the inverse of covariance matrix. Thus Ledoit & Wolf [10] have used Frobenius norm to measure the quadratic distance between the estimated and the true covariance matrix. Under the assumption that n observations are fixed while p features tends to infinity, they prove that the optimal value for shrinkage intensity, α asymptotically behaves like a constant p . This constant, k , is explained as followed:

$$k = \frac{\pi - \chi}{\gamma} \quad (5)$$

The π, χ and γ are known as consistent estimator parameters to calculate optimal intensity. Refer to [10] for extensive explanation regarding those estimators. In mass spectrometry analysis, feature extraction plays a vital role in extracting discriminant features from the potential peaks signal. Shrinkage covariance estimation could be seen as potential techniques for assembling and calibrating peaks into peaks-bin or m/z windows due to well-structured estimation in predicting correlations among peaks for high-dimensionality of data. Hence, it produces well discriminant and independent features for feature selection process.

3 Artificial Bee Colony (ABC) as Feature Selection

Artificial Bee Colony is derived from the foraging behaviour and consists of three different agents that play roles in searching for quality nectar and improve their survival in the population. ABC algorithm as feature selection is modelled based-on previous study [20,18] by removing and modifying some original parameter settings proposed by [8] to suit the initial data of ovarian cancer dataset gathered from peaks extraction phase. In general, the ABC algorithm as feature selection follows the foraging behaviour of employee bees, onlooker bees and scout bees; meanwhile neighbourhood search, parameter setting and evaluation criteria have been modified to suit the data. A brief discussion of ABC as feature selection is presented as follow:

Initial Population: Initial population is constructed depends on problem to solve. It represents search space for the algorithm to explore and optimise its findings. A search space is denoted as $X_{i,d}$ where i is the number of employee bees, meanwhile d is the number of solution to be optimised by particular bees. Each element of d must be a unique food source.

Neighbourhood Search: Each employee bees will iterate to improve their initial nectar amount by generating new food source, $V_{i,d}$. Exploiting new search space will require the neighbourhood search mechanism to randomly modify any $d = 1, 2, \dots, D$ from particular X_i , where D is maximum solution for each particular bee. Thus, to comprehend with initial data in this study, a simple random search is denoted as follow to produce better exploitation result.

$$V_{i,d} = X_{a,b} \quad (6)$$

Where i and d could be any $i = 1, 2, \dots, SN$ and $d = 1, 2, \dots, D$ respectively, whilst SN is maximum number of employee bees. Apart from that, $X_{a,b}$ must be different from any $V_{i,d}$ for particular i . The nectar amount of new food source will be evaluated and compared with the current one. It will then be replaced by the new one if the evaluation shows the new nectar or fitness value is superior and vice versa.

Solution Score Evaluation: We follow the original ABC algorithm that improves the searching process by decreasing their objective function, $objVal_i$ prior probabilistic selection. Therefore, objective function is constructed by minimizing classification error of linear SVM. Then, fitness fit_i or nectar quality of the associated food sources is measured as follow:

$$fit_i = \frac{1}{1 + objVal_i} \quad (7)$$

Probabilistic Selection: On the second phase of foraging, all employee bees meet the onlooker bees on the waggle dance to disseminate all the best so far nectars quality. Onlooker bees will only evaluate and select food sources that meet the criteria of probability P_i as follow for further process.

$$P_i = \frac{fit_i}{(\sum_i^{SN} fit_i)} \quad (8)$$

Exhaustive Search: The performance of foraging behaviour is controlled by a predetermined number of cycles called limits. Limits are assigned as surveillance for every employee bees to identify the exhausted search that might happen to particular bees. Therefore, when a food source cannot be improved further (exhausted), the associate employee bee will be changed to scout bee. This new scout bee will do new exploration to the food sources in the search space.

4 Implementation and Discussion

This study makes use of ovarian dataset downloaded from National Cancer Institute (home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp). This high-dimensional ovarian cancer dataset contains 216 samples comprises 121 of cancer samples and 95 of normal samples. In general, mass spectrometry analysis involves several steps of consecutive processes as discussed in [20]. Hence this analysis starts with some common pre-processing methods, followed by feature extraction process, feature selection and classification prior to biomarker test.

This study has applied baseline removal, normalisation and noise filtering in the pre-processing step consecutively. We retain only positive values by utilising the morphological operator of *imtophat* in matlab function for the baseline correction. Meanwhile, method proposed by [1] is applied for normalisation and followed by matlab function of wavelet *mssgolay* for noise filtering. After these three common pre-processing methods, peaks locations are identified across each spectrum. Again, we apply Armananzas et al. [1] approach that utilises peak detection method from several previous studies.

In order to extract potential peaks as well-discriminated and independent features, shrinkage covariance estimator is applied to calibrate and assemble peaks into group of peaks-bin (also known as m/z windows). Peaks-bin are constructed in terms of strong correlation among its neighbourhood where shrinkage covariance estimation is converted into correlation and repeatedly evaluated onto all peaks until no more strong correlations occurred. We consider strong correlation when the value of correlation among features exceeds 0.80. The proposed shrinkage covariance estimation has been elaborated in section two. Instead of shrinkage, we also apply empirical covariance correlation to the ovarian data that follow the same pre-processing steps. The first 400 features are selected from the whole 216 samples after ranking them based on frequently being chosen across samples by following the same steps as collecting features from shrinkage approach. These features are split into training and testing samples for both disease and control cases in ratio 70:30 respectively.

Feature selection was applied to select the most parsimonious features to be classified as biomarkers. ABC algorithm was then applied to select feature subset from 400 most prominent features. The algorithm starts by initialising all 400 features randomly among 50 numbers of employee bees, $i = 1, 2, \dots, SN$. Where equally 8 different features, d is assigned for every agent. These features are known as food sources and each agent is responsible to optimise her food sources based on their nectar quality. On this stage, ABC algorithm will incorporate linear SVM classifier with 10 k-folds to evaluate the fitness value or nectar

quality of each food source. In the ABC algorithm only employee bees have the direct contact with all the available food sources, thus capable to produce mutant solution by randomly choosing new food source through neighbourhood search in equation 6. New combination of food source will be evaluated through objective function from equation 7 and produce new nectar quality from equation 8. Hereby, the new solution will replace the current one if and only if the produce solution is better.

For feature selection purposes, we tune the parameter setting for the colony ratio in between employee bees and onlooker bees. According to the original setting, both colonies have the same number. We have tested the original setting and our setting ratio, 50:50 and 50:100 respectively on liver datasets [18] and ovarian dataset in current study. Anyhow, we find that our new setting ratio 50:100 exploits better food sources with stopping criteria is 100 cycles and limits is 100. Onlooker bees improve the optimisation of food sources passed by transient employee bees and select only those have good probability evaluation as in equation 8. In addition, onlooker bees also perform modification to the selected food source position correspond to neighbourhood search and nectar quality evaluation through equation 6 and 7. The foraging process is repeated for at least 100 times or will stop earlier if classification error approaching 0. We set maximum cycle (MCN) or stopping criteria as 100. All the parameters setting are shown in table 1 and based-on the best performance from several previous studies of population based-algorithm. Details of algorithm and implementation ABC as feature selection are referred to [18].

Table 1. Parameter setting for ABC implementation

Population size	50
Employee bees	50
Onlooker bees	100
limit	100
MCN	100

The results generated from the training dataset is analysed after 100 runs and evaluated onto testing datasets. The analysis from two different feature extraction methods are analysed and evaluated separately. Table 2 shows the comparison from both shrinkage and empirical covariance correlation methods that is based on evaluation of most occurrence features. Both methods perform well for training data, anyhow slightly different on testing data. Accuracy measure the veracity of the diagnostic test from classification model, sensitivity is proportional to the true diagnosis of cancer cases meanwhile specificity is proportional to the true diagnosis of normal cases. Empirical covariance-correlation method shows better prediction on sensitivity; meanwhile shrinkage covariance-correlation performs better results for specificity.

We emphasize on finding well discriminant features as our objective of the study. These features need to allow a distinction between cancer with normal

Table 2. The eight most occurrence m/z values

	Shrinkage covariance-correlation		Empirical covariance-correlation	
	training	testing	training	testing
Accuracy	1	0.9531	1	0.9531
Sensitivity	1	0.9630	1	1
Specificity	1	0.9459	1	0.9231
m/z windows	7038.264-7047.627		3860.689-3863.356	
	7050.509-7055.554		3893.821-3895.963	
	7067.093-7070.701		4299.631-4304.698	
	7721.58-7736.608		5129.705-5135.238	
	7923.418-7925.71		7049.789-7054.833	
	8692.148-8692.948		7721.518-7734.343	
	8704.954-8716.969		7923.418-7925.71	
	10323.28-10324.15		8931.409-8938.71	

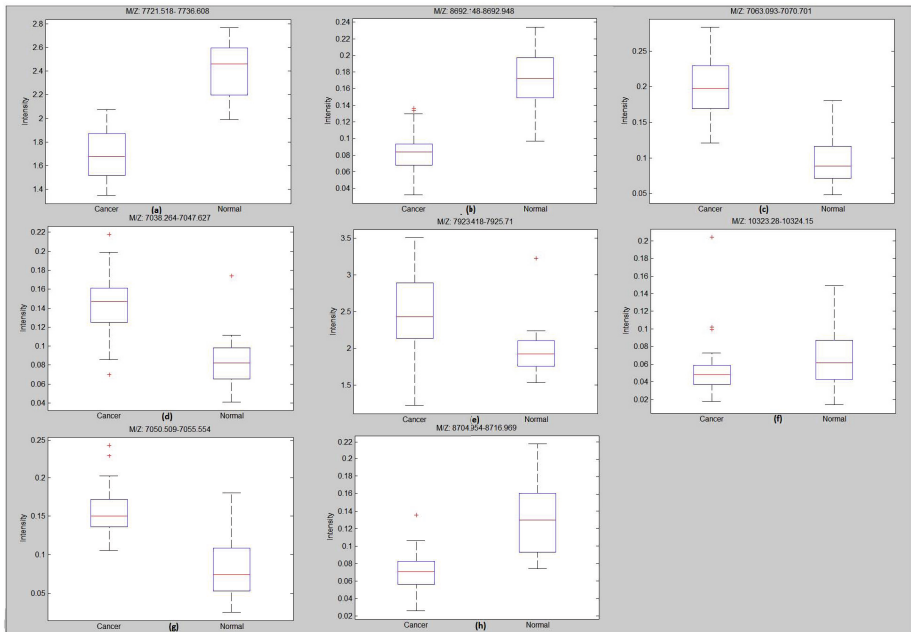


Fig. 1. Box plot of eight most potential markers from shrinkage covariance correlation

cases, which required us to draw the box plot representation for eight of most selected features from both analyses. According to Massart and Smeyers-verbeké [13], classical statistical methods such as F-test, t-test and analysis of variances (ANOVA) are normal distribution oriented assumption and vulnerable when the data contains outliers. On the other hand, box plot technique is excellent in representing differences between datasets without any statistical assumption. Figure 4

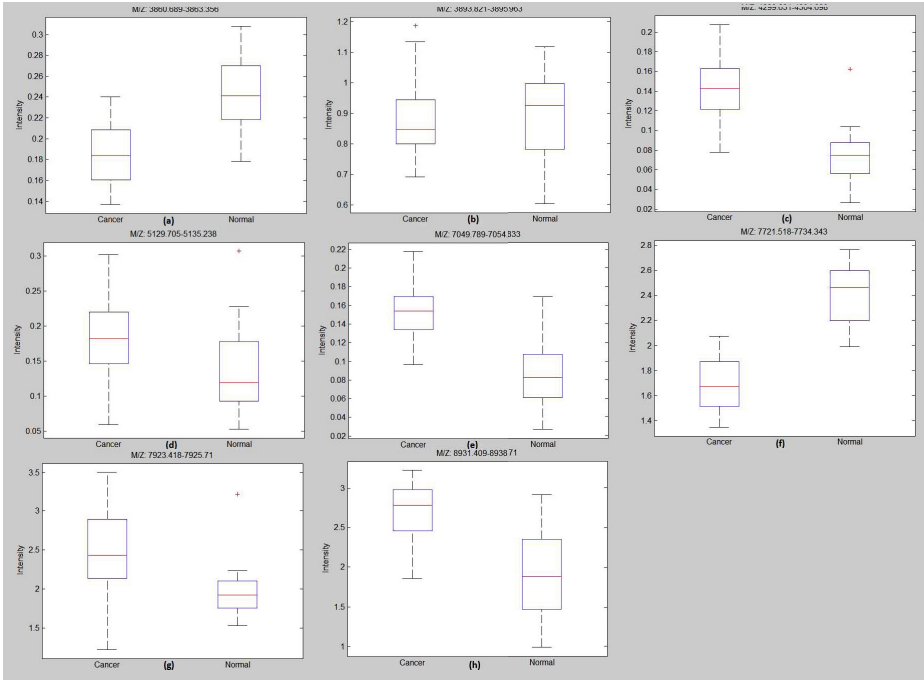


Fig. 2. Box plot of eight most potential markers from empirical covariance correlation

depicts eight most potential markers represented from shrinkage covariance-correlation analysis, meanwhile figure 4 for empirical covariance-correlation.

In general, all potential features selected by shrinkage covariance correlation in figure 4 shows significant discrimination between cancer and normal samples thus prove usefulness as markers. Features *a*, *b*, *c*, *d*, *e*, *g* and *h* discriminate most of the cancer and normal samples, meanwhile cancer sample from feature *f* in some extend separate from normal cases. In figure 4, features *b* and *d* are not able to perform discrimination between cancer and normal cases. Meanwhile both features *f* and *g* are also been selected as potential markers for shrinkage. Anyhow from both shrinkage covariance-correlation and empirical covariance correlation, there is no features completely discriminant between cancer and normal cases.

5 Conclusion

In this paper we showed that the use of shrinkage estimator for covariance and correlation is much better than empirical correlation method in interpreting biological insight of mass spectrometry data in which the number of features are much bigger than number of samples $n \ll p$. This method is applied to assemble and calibrate detected peaks that have strong correlation and extract only the

most discriminant peaks-bin or m/z windows for further process. Furthermore, incorporating Artificial Bee Colony (ABC) as feature selection and linear SVM as classifier yields good classification performance in identifying potential markers. This study will be further extended to explore both shrinkage estimator for feature extraction and ABC algorithm as feature selection for other types of mass spectrometry datasets. The algorithm will be compared with other types of metaheuristic algorithm and incorporated with other types of classifiers to predict biomarkers for particular diseases.

Acknowledgement. We wish to express our heartfelt gratitude to the Ministry of Higher Education (MOHE) for funded grant under Fundamental Research Grant Scheme (FRGS) (203/PKOMP/6711268). Lastly, the first author would like to thank Universiti Teknologi MARA (UiTM) for offering a generous Ph.D. scholarship.

References

1. Armananzas, R., Saeys, Y., Inza, I., Garcia-Torres, M., Bielza, C., Van de Peer, Y., Larranaga, P.: Peakbin selection in mass spectrometry data using a consensus approach with estimation of distribution algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 8(3), 760–774 (2011)
2. Celik, M., Karaboga, D., Koylu, F.: Artificial bee colony data miner (abc-miner). pp. 96–100. *IEEE* (2011)
3. Celis, J.E., Gromov, P.: Proteomics in translational cancer research: toward an integrated approach. *Cancer Cell* 3(1), 9–15 (2003)
4. Coombes, K.R., Tsavachidis, S., Morris, J.S., Baggerly, K.A., Hung, M.C., Kuerer, H.M.: Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform. *Proteomics* 5(16), 4107–4117 (2005)
5. Efron, B., Morris, C.: Data analysis using stein’s estimator and its generalizations. *Journal of the American Statistical Association* 70(350), 311–319 (1975)
6. He, Z., Yu, W.: Stable feature selection for biomarker discovery. *arXiv preprint arXiv:1001.0887* (2010)
7. James, W., Stein, C.: Estimation with quadratic loss. In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 361–379 (1961)
8. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. *Techn. Rep. TR06*, Erciyes Univ. Press, Erciyes (2005)
9. Karaboga, D., Ozturk, C.: A novel clustering approach: Artificial bee colony (abc) algorithm. *Applied Soft Computing* 11(1), 652–657 (2011)
10. Ledoit, O., Wolf, M.: Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance* 10(5), 603–621 (2003)
11. Ledoit, O., Wolf, M.: A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis* 88(2), 365–411 (2004)

12. Listgarten, J., Emili, A.: Statistical and computational methods for comparative proteomic profiling using liquid chromatography-tandem mass spectrometry. *Molecular & Cellular Proteomics* 4(4), 419–434 (2005)
13. Massart, D.L., Smeyers-Verbeke, A.J.: *Practical Data Handling Visual Presentation of Data by Means of Box Plots* (2005)
14. Mohd Shukran, M.A., Chung, Y.Y., Yeh, W.C., Wahid, N., Ahmad Zaidi, A.M.: Artificial bee colony based data mining algorithms for classification tasks. *Modern Applied Science* 5(4), 217 (2011)
15. Resson, H.W., Varghese, R.S., Drake, S.K., Hortin, G.L., Abdel-Hamid, M., Lofredo, C.A., Goldman, R.: Peak selection from maldi-tof mass spectra using ant colony optimization. *Bioinformatics* 23(5), 619–626 (2007)
16. Sanavia, T., Aioli, F., Da San Martino, G., Bisognin, A., Di Camillo, B.: Improving biomarker list stability by integration of biological knowledge in the learning process. *BMC Bioinformatics* 13(suppl. 4), S22 (2012)
17. Schäfer, J., Strimmer, K., et al.: A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology* 4(1), 32 (2005)
18. SyarifahAdilah, M., Abdullah, R., Venkat, I.: Abc algorithm as feature selection for biomarker discovery in mass spectrometry analysis. In: *2012 4th Conference on Data Mining and Optimization (DMO)*, pp. 67–72. IEEE (2012)
19. Yao, J., Chang, C., Salmi, M., Hung, Y., Loraine, A., Roux, S.: Genome-scale cluster analysis of replicated microarrays using shrinkage correlation coefficient. *BMC Bioinformatics* 9(1), 288 (2008)
20. Yusoff, S.A.M., Venkat, I., Yusof, U.K., Abdullah, R.: Bio-inspired metaheuristic optimization algorithms for biomarker identification in mass spectrometry analysis. *International Journal of Natural Computing Research (IJNCR)* 3(2), 64–85 (2012)

Metaoptimization of Differential Evolution by Using Productions of Low-Number of Cycles: The Fitting of Rotation Curves of Spiral Galaxies as Case Study

Miguel Cárdenas-Montes¹, Miguel Á. Vega-Rodríguez², and Mercedes Mollá¹

¹ Centro de Investigaciones Energéticas Medioambientales y Tecnológicas,
Department of Fundamental Research, Madrid, Spain
{miguel.cardenas,mercedes.molla}@ciemat.es

² University of Extremadura, ARCO Research Group,
Dept. Technologies of Computers and Communications, Cáceres, Spain
mavega@unex.es

Abstract. In order to increase the efficiency of Evolutionary Algorithms, practitioners include improvements as new operators or modifications of the canonical operators, or the hybridization with other Evolutionary Algorithms. However, an alternative to obtain high-quality solutions is: to tune the parameters which govern the behaviour of the algorithm to the specific problem to optimize. This parameters adjustment can be performed by using other Evolutionary Algorithm (Metaoptimization). Unfortunately, metaoptimization leads to a critical increment in the execution time. In this work, a measure of the quality of the tuned behavioural parameters when executing very low-number of cycles in the optimizer is performed and compared with the case when executing high-number of cycles. The fundamental aspect of this approach is if there is enough information about the quality of the behavioural parameters in the very initial cycles of the optimizer. By ascertaining if productions based on a low-number of cycles harvest high-quality behavioural parameters, one of the main drawbacks of the metaoptimization process —the large execution time— can be overcome. The performed tests —the fitting of experimental data of rotation curves of spiral galaxies— demonstrate that this approach improves the efficiency of the metaoptimizer, while reducing processing time.

Keywords: Metaoptimization, Differential Evolution, Rotation Curve, Spiral Galaxy.

1 Introduction

During the development of metaheuristic techniques, the optimizers require to fix the values of diverse behavioural parameters. In general, these parameters govern the behaviour of the algorithms, and therefore, they are key elements in its final efficiency.

In the past, approaches based on the *factorial design* have been followed to optimize the behavioural parameters. However, this procedure oversimplifies the problem, neglecting the potential relationships between the behavioural parameters. Neither, by-hand selection of the most suitable set of parameters is an affordable task.

As any other complex problem, the adjustment of the behavioural parameters of an evolutionary algorithm can be treated by other evolutionary algorithm, termed *metaoptimizer* or *tuner*. This kind of optimization is termed *metaoptimization*.

Unfortunately, the metaoptimization carries out a relevant increment of the execution time. If the problem to optimize takes long, or a high-number of cycles or large population are required to obtain high-quality solutions, then the scenario aggravates. Therefore, it is necessary to evaluate if a lower number of cycles in the optimizer produces behavioural parameters of enough quality for the problem, and consequently, processing time can be saved; although this low-number of cycles of the optimizer is not producing so-high-quality solutions.

If the behavioural parameters used in the optimizer (algorithm which is optimized) exhibit its quality from the very initial cycles, then large executions can be avoided, as well as the number of cycles truncates before the optimizer reaches its stagnation level. Moreover, the number of cycles of the optimizer is a control mechanism over the execution time budget and, indirectly over the quality of the solutions of tuner.

Particularly, this work focuses on tuning the behavioural parameters of Differential Evolution (DE) algorithm [1,2]. This election is based on the popularity of the algorithm, frequently used in optimization in artificial functions and real-world problems. Concerning the proposed problem, the adjustment of experimental data set to a theoretical curve—rotation curve of spiral galaxy—is used as benchmark.

The rotation curve of a galaxy is defined as the relationship between the rotational velocity of stars as function of the radial distance to the galaxy centre. The relevance of this problem stems from the discrepancy between the observed velocity of the stars and the Newtonian-Keplerian prediction, in such way that masses derived from the rotational kinematics and gravitational laws do not match. Nowadays, this discrepancy is explained by the presence of dark matter, which is not emitting light. As a consequence, the characterization of rotation curve in spiral galaxies is a measure of the amount of dark matter in the galaxy.

Dark energy and dark matter have never directly been observed, and their nature remains unknown. Understanding the nature of the dark matter and the dark energy is one of the most important challenges of the current cosmology studies¹.

The experimental data sets correspond to the orbital velocity of stars for spiral galaxies: NGC 2460 and NGC 3370 [3]. Due to the inherent experimental error of data, the data volume and the fact that both arms of the galaxies do not exhibit the same velocity curve; this fitting process becomes challenging, allowing many almost-equal sub-optimal adjustments.

The rest of the paper is organized as follows: Section 2 summarizes the Related Work and previous efforts done. In Section 3.1, the most relevant details about the implementation are presented. In Section 3.2, the physical problem is briefly described. The underpinning of the Statistical Inference is exposed in Section 3.3. The Results and the Analysis are displayed in Section 4. Finally, the Conclusions and the Future Work are presented in Section 5.

¹ The quantification of the budget between ordinary and dark components in the Universe is a major issue as proven by the recognition of the Science magazine in 1998 and 2003 as *Scientific Breakthrough of the Year*.

2 Related Work

Diverse works have examined aspects of parameter tuning in Evolutionary Algorithms. Early in the bibliography, the drawback associated with the large execution time is reported. In one of the pioneer studies in metaoptimization [4], it was already stated the large computational cost as limiting factor in metaoptimization processes.

A very popular strategy to overcome the large execution times of metaoptimization is *Racing* [5]. The aim of this strategy is to reduce the number of tests to estimate the utility of a behavioural parameters set. After an initial phase where all sets are equally estimated, the algorithm separates good and poor configurations, for later focussing on good ones by incrementing their number of evaluations.

The Racing strategy has suffered from modifications aimed to accelerate the discrimination of poor solutions. The variants differ on the criteria used to sift the poor configurations. For example, it can be mentioned: the use of a Gaussian distribution centred at the current best candidate to generate the next generation [6]; or *F-Race* where the Friedman test is used to promote or discard the candidates into the next iteration. *F-Race* has been applied to Ant Colony Optimization for traveling salesman problem [7] and to *iterated local search* and *simulated annealing* for timetabling problem [8].

Other attempt of DE metaoptimization is presented at [9]. In this work, a suite of twelve fitness functions (separable and non-separable) are used as benchmark. The differences emerge in the general approach of the problem. In [9] the metaoptimization of DE is monolithic for the whole suite: the behavioural parameters are tuned for the suite; whereas in our work each case is treated independently. Finally, this work also underlines the disadvantage associated to the large execution time when evaluating the benchmark suite for the highest dimensionality (100 dimensions).

Finally, a review of the approaches for tuning the behavioural parameters of metaheuristics is presented in [8]. The review begins with the drawbacks of the *trial-and-error* approach, passing by a methodology based on *factorial design*; and finishing with *F-Race* approach. The time-consumption disadvantage when applying metaoptimization to industrial problems is also underlined. Other review of methods for parameter tuning can be found at [10]. Unfortunately, this work focusses only on one single separable function (Rastrigin function), which prevents any comparison process.

Our approach proposes to study the quality solutions obtained when evaluating behavioural parameters with low-number of cycles in the optimizer and to compare them with the high-number of cycles. To the authors' knowledge, up to now, this approach has not been addressed in the past.

3 Methodology

3.1 Implementation

In order to deal with a whole evolutionary algorithm, a python implementation is proposed for the tuner. Python election is based on its capacity to handle pieces of text, to compose files with these pieces, then, to compile the source code, to execute it and to capture output information from the execution. By repeating this process, the behavioural parameters of the optimizer can evolve. In our work, both tuner and optimizer implement DE algorithm.

On the other hand, the evolutionary algorithm which parameters are being optimized is codified in C language. C language election is based on the need of a fast execution for the problem under optimization. Additionally to the cited benefit, this different codification eases the identification of each part of the code while codifying.

One of the critical points of the metaoptimizer part is to capture the final fitness of the evolutionary algorithm to be recorded as the fitness of the metaoptimizer individual. For this, the best fitness is recorded in a text file after executing the problem and captured by python from this file. So, synchronization operations during the writing and reading are required.

Both, tuner and optimizer Differential Evolution [1,2] have been implemented with the schema DE/rand/1/bin [11]. Furthermore, in all numerical experiments, the configuration in the tuner is a population of 10 vectors and 10 cycles. Otherwise, in the optimizer, the population is composed by 10 vectors; and two configurations for the number of cycles: 10 and 1,000. In all cases, real-valued representation is used. The behavioural parameters of the tuner are fixed with values $\mu = CR = 0.5$.

As pseudorandom number generator, a subroutine based on Mersenne Twister [12] has been used in both implementations: python and C.

The numerical experiments are executed in a single core of a computer with two Intel Xeon X5570 processors at 2.93 GHz and 8 GB of RAM. The C code has been compiled by using gcc version 4.4.5 with optimization level -O3.

3.2 The Rotation Curve in Spiral Galaxies

The proposed problem corresponds to the adjustment of experimental data —orbital velocity of stars in spiral galaxies (Fig. 1)— to a theoretical curve. The election of this problem resides on the difficulty to fit data with experimental errors, the fact that two arms of spiral galaxies which usually do not overlap. The curves have been extracted from a larger astronomical data set, covering approximately 56 galaxies [3]. The selection criterion has been the two largest populated data set.

This problem has been used in the past in optimization problems: as benchmark function in numerical optimization in order to study the sensitiveness of evolutionary algorithms to the choice of the random number generator [13], and it has been also treated from the physical point of view [14,15].

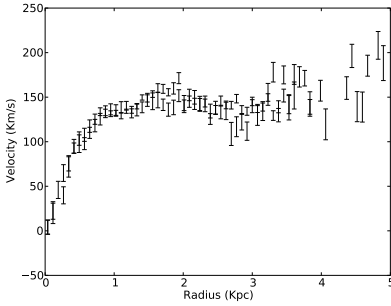
The velocity of the stars is characterized by an equation with physical meaning describing the four mass contributions to the rotation curve —bulge, disk, interstellar gas and halo— (Eq. 1).

$$v^2(r) = v_D^2(r) + v_B^2(r) + v_H^2(r) + v_G^2(r) \tag{1}$$

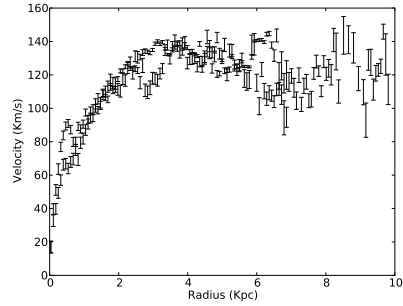
Except for the halo, the other three contributions are merged in a variable, whereas the halo contribution is modelled by Eq. 2. Therefore, the number of parameters to adjust is three: $v_D^2 + v_B^2 + v_G^2$, σ , and α .

$$v_H^2(r) = 2 \cdot \sigma^2 \cdot \left(1 - \left(\frac{r}{\alpha}\right) \cdot \text{tang}^{-1}\left(\frac{\alpha}{r}\right)\right) \tag{2}$$

$$\chi^2 = \sum_{\text{for all points}} \left(\frac{y_{\text{simulated}} - y_{\text{observed}}}{\text{Error}_{\text{observed}}}\right)^2 \tag{3}$$



(a) Rotational Velocity for NGC 2460



(b) Rotational Velocity for NGC 3370

Fig. 1. Rotational velocities and errors of stars in galaxies: NGC 2460 and NGC 3370 employed as experimental data set

According to the usual practice in adjustment of experimental data to theoretical curve, the chi-squared test has been chosen as fitness function (Eq. 3). The lower the χ^2 is, the closer the solution is to the objective —the fitter the experimental data is to theoretical curve.

3.3 Statistical Inference

In this work, the usual statistical analysis in the numerical optimization works has been followed [16,17]. The analysis is based on non-parametric tests, such as: Kruskal-Wallis and Wilcoxon signed-rank tests. Non-parametric tests have been selected because they do not assume any explicit condition on the data, for example normality. In-depth description of the statistical tests is beyond of the scope of this paper.

4 Results and Analysis

In order to check the hypothesis of the capacity of the tuner to produce competitive behavioural parameters by using a reduced number of cycles (10) in the optimizer, a production composed of 25 executions is performed per case. Later, these behavioural parameters are compared with the behavioural parameters emerged from a production with high-number of cycles (1,000). The aim of this experiment is to compare the similarities and differences of the behavioural parameters obtained (Fig. 2).

On the other hand, a statistical analysis of the numerical results when using behavioural parameters tuned with low-number of cycles and high-number of cycles is performed (Table 1). And finally, once the efficiency of the approach proposed has been verified, the processing time saved is presented (Table 2).

Metaoptimization Production. After each execution of DE tuner, a couple of values (μ , CR) are obtained as tuned behavioural parameters for the problem under optimization (Fig. 2). Additionally to the scatter plot $\mu - CR$, at top and at right of each figure, the histograms with the frequency of each value are also plotted.

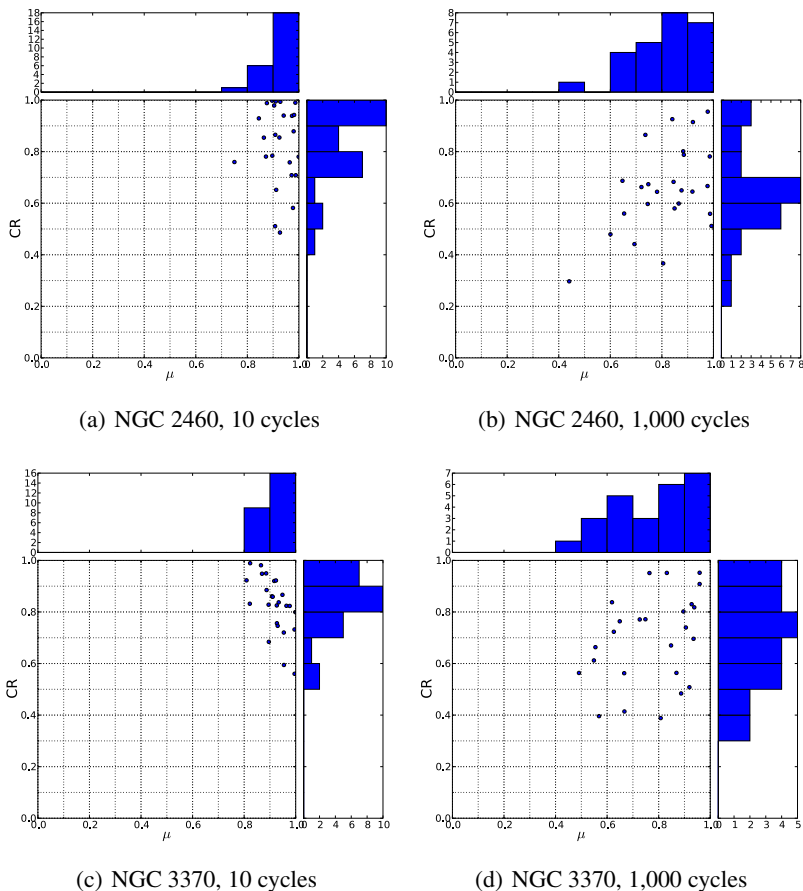


Fig. 2. Results (μ and CR) of metaoptimizer after 25 executions for galaxies: NGC 2460 and NGC 3370, and for 10 and 1,000 cycles. Top and right: the histogram of each parameter.

As can be appreciated, most of the tuned parameters are located in the upper-right quadrant. These results are slightly different, but congruent with the recommendation ($\mu = 0.8$ and $CR = 0.9$) of Prof. Storn² for the schema DE/rand/1/bin; although the original recommendation is stated for artificial separable functions. Next, the mid-point of the most populated division is employed to decide about the most suitable behavioural parameters (μ , CR) for the problem to optimize (Table 1).

Following the procedure established, the metaoptimizer tunes the behavioural parameters of optimizer using two configurations (10 and 1,000 cycles). This produces two sets of suitable behavioural parameters per galaxy (Fig. 2).

By observing the values obtained for μ and CR , it is appreciated the similarities in the values, independently of the number of cycles of the optimizer. This reinforces the hypothesis that the quality of the behavioural parameters can be extracted from the few

² <http://www1.icsi.berkeley.edu/~storn/code.html>

Table 1. Best fitness (25 executions) for each galaxy and case. The numerical results labeled with: *random* have been obtained with $\mu = CR = 0.5$, those labeled with *optimized* by using μ and CR optimized with 10 or with 1,000 cycles. The numerical results without label correspond to the cases where μ and CR have been optimized with 10 cycles and the runs executed with 1,000 cycles.

Galaxy	μ	CR	Cycles	Mean fitness	Comment	Statistical Test (p-value)
NGC 2460	0.50	0.50	10	$57,518.6 \pm 16,848.4$	Random	Wilcoxon signed-rank $1.2 \cdot 10^{-5}$
	0.95	0.95	10	$2,938.3 \pm 1,966.7$	Optimized	
	0.50	0.50	1,000	$1,247.4 \pm 630.3$	Random	Kruskal-Wallis $2.8 \cdot 10^{-12}$
	0.95	0.95	1,000	$314.5 \pm 1.14e-13$		
	0.75	0.65	1,000	375.2 ± 222.1	Optimized	
	NGC 3370	0.50	0.50	10	$353,444.0 \pm 61,195.0$	Random
0.95		0.85	10	$28,741.8 \pm 16,472.1$	Optimized	
0.50		0.50	1,000	$11,613.7 \pm 6,472.7$	Random	Kruskal-Wallis $9.7 \cdot 10^{-14}$
0.95		0.85	1,000	$2,873.9 \pm 4e-13$		
0.95		0.75	1,000	$2,873.9 \pm 4e-13$	Optimized	

initial cycles. The most suitable values for NGC 2460 and 10 cycles are $\mu = CR = 0.95$ (Fig. 2(a)), whereas for 1,000 cycles are $\mu = 0.75$ and $CR = 0.65$ (Fig. 2(b)). For the galaxy NGC 3370, the most suitable values for low-number of cycles are $\mu = 0.95$ and $CR = 0.85$ (Fig. 2(c)), whereas for high-number of cycles³ are $\mu = 0.95$ and $CR = 0.75$ (Fig. 2(d)).

The next step is to verify if the efficiency of each set is significantly different.

Fitness Analysis. In order to discriminate if the tuned behavioural parameters of DE are more efficient when the tuning process has been performed with 10 or with 1,000 cycles; 25 runs of the optimizer are executed per case (Table 1).

Concerning the numerical results for the galaxy NGC 2460, it can be observed that the tuned parameters with low-number of cycles ($\mu = CR = 0.95$) outperform the tuned parameters with high-number of cycles ($\mu = 0.75$, $CR = 0.65$) when both executing 1,000 cycles. For the galaxy NGC 3370, the same comparison leads to both cases: low-number ($\mu = 0.95$, $CR = 0.85$) and high-number ($\mu = 0.95$, $CR = 0.75$) of cycles produce the identical mean fitness. As expected, whatever tuned behavioural parameters, independently of the number of cycles, outperform randomly selected behavioural parameters ($\mu = CR = 0.5$).

From the proposed experimental setup and the numerical results, it can be concluded that a reduction in the number of cycles of optimizer, at least, does not degrade the quality of the behavioural parameters obtained in the metaoptimization process. Based only on the initial cycles of the optimizer, the tuner is able to capture enough information about the quality of the behavioural parameters to evaluate them.

³ In the previous cases —galaxy and number of cycles— the mid-point of the most populated division is selected to establish the most suitable values of μ and CR . However for the galaxy NGC 3370 and 1,000 cycles configuration, neither division is populated with more than 2 points. Therefore, the most populated bin in the histogram is used as criterion to select the suitable behavioural parameters.

Statistical Analysis. In order to check if the differences in the fitness (Table 1), when using behavioural parameters tuned with low-number and high-number of cycles in the optimizer, are significant, the production 25 executions is statistically analysed.

The statistical analysis of data is performed by using Kruskal-Wallis test for multiple comparisons, Wilcoxon signed-rank test for pair comparison and finally, sign test to discern if the optimized set of parameters outperforms or not the standard ones. In all cases, non-parametric tests have been chosen because they do not require explicit conditions for data distribution.

Except for the case of NGC 3370, 1,000 cycles and the two sets of tuned behavioural parameters —where identical numerical results are obtained—, the Kruskal-Wallis and Wilcoxon signed-rank tests indicate that the differences for the numerical results are significant for a confidence level of 95% (p-value under 0.05). This means that the differences are unlikely to have occurred by chance with a probability of 95%.

Execution Time. In the previous points, the analysis focussed on the values achieved for the tuned behavioural parameters of DE and on the numerical results obtained with these values. It has been proved that metaoptimization based on optimization process with low-number of cycles can produce high-quality behavioural parameters for DE algorithm. Once the numerical efficiency of the approach has been checked, the corresponding processing times are presented (Table 2).

Table 2. Mean execution time of both tuner and optimizer for 10 and 1,000 cycles in the optimizer

Galaxy	Cycles	Execution Time	Cycles	Execution Time	Reduction
Optimizer					
NGC 2460	10	6.56 ms	1,000	412.24 ms	98.4%
NGC 3370	10	12.24 ms	1,000	654.12 ms	98.1%
Tuner					
NGC 2460	10	17.41 s	1,000	86.58 s	79.9%
NGC 3370	10	18.23 s	1,000	146.10 s	87.5%

As can be appreciated, Table 2 shows a significant reduction of the execution times for both: tuner and optimizer when a low-number of cycles are employed in the optimizer. The execution time reduction is higher than 98% for the optimizer, while ranging from 79.9% to 87.5% for the tuner.

Through optimizing behavioural parameters in this scenario, a saving of processing time is achieved, at the same time that high-quality solutions are produced. This is specially relevant for industrial applications where execution time is as relevant as the fitness; and for the cases where an optimization process is applied successively to different data sets. By varying the number of cycles in the optimizer, the metaoptimization process is endowed of a control over the quality of the achieved solutions and over the processing time budget.

5 Conclusions and Future Work

In this paper, an approach to tune the behavioural parameters of Differential Evolution algorithm, and simultaneously saving processing time has been proposed. This approach

measures the quality of the tuned behavioural parameters when a low-number of cycles is applied to the optimization process in opposition to when the optimization process is performed with high-number of cycles. The success of this approach holds in the capacity of tuner to evaluate the quality of the behavioural parameters from the very initial cycles. By implementing this approach, a strong reduction of the execution time is achieved while maintaining high-quality behavioural parameters. The proposed approach has been initially applied to the fitting of rotation curves of spiral galaxies.

Finally, the numerical results and the later analysis demonstrate that the proposed approach reduces the execution time while delivering high-quality behavioural parameters. Furthermore, it allows a finer control over the global processing time and, at the same time, over the quality of the solutions.

More comparative works, where the proposed technique is applied to other complex problems, other schemas different from the DE/rand/1/bin, and confronted to other approaches: Race or F-Race, are considered as Future Work. As candidates, the optimization of non-separable functions in high-dimensional problems, and problems in Astrophysics area can be cited. This kind of functions are candidates by the difficulty to find high-quality solutions. Among the candidates, the following functions are proposed: Schaffer F6 and F7, Schwefel Problem 1.2, Rana, and Rosenbrock. Furthermore, potential variants of the approach are also considered.

Acknowledgement. This work has been partially supported by DGICYT grant AYA2010–21887–C04–02. Also, by the Comunidad de Madrid under grant CAM S2009/ESP-1496 (AstroMadrid) and by the Spanish MICINN under the Consolider-Ingenio 2010 Program grant CSD2006-00070: First Science with the GTC (<http://www.iac.es/consolider-ingenio-gtc>) which are acknowledged. We would like to thank Dr. Isabel Márquez warmly for sending the tables associated to the rotation curves used in this work. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) via the project EGI-InSPIRE under the grant agreement number RI-261323.

References

1. Storn, R., Price, K.V.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization* 11(4), 341–359 (1997)
2. Price, K.V., Storn, R., Lampinen, J.: *Differential Evolution: A practical Approach to Global Optimization*. Springer, Berlin (2005)
3. Marquez, I., et al.: Rotation curves and metallicity gradients from HII regions in spiral galaxies. *Astron. Astrophys.* 393, 389–408 (2002)
4. Mercer, R., Sampson, J.: Adaptive search using a reproductive metaplan. *Kybernetes* 7, 215–228 (1978)
5. Maron, O., Moore, A.W.: The racing algorithm: Model selection for lazy learners. *Artif. Intell. Rev.* 11(1-5), 193–225 (1997)
6. Yuan, B., Gallagher, M.: Combining Meta-EAs and Racing for Difficult EA Parameter Tuning Tasks. In: Lobo, F.G., Lima, C.F., Michalewicz, Z. (eds.) *Parameter Setting in Evolutionary Algorithms*. SCI, vol. 54, pp. 121–142. Springer, Heidelberg (2007)
7. Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K.: A racing algorithm for configuring metaheuristics. In: *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, New York, USA, July 9-13, pp. 11–18. Morgan Kaufmann (2002)

8. Birattari, M.: Tuning Metaheuristics. *SCI*, vol. 197. Springer, Heidelberg (2009)
9. Pedersen, M.E.H.: Good Parameters for Differential Evolution. Technical Report Technical report no. HL1002, Hvass Laboratories, University of Zurich, Department of Informatics (2010)
10. Smit, S.K., Eiben, A.E.: Comparing Parameter Tuning Methods for Evolutionary Algorithms. In: *IEEE Congress on Evolutionary Computation (CEC)*, pp. 399–406 (May 2009)
11. Mezura-Montes, E., Velázquez-Reyes, J., Coello, C.A.C.: A comparative study of differential evolution variants for global optimization. In: *GECCO, Genetic and Evolutionary Computation Conference*, Seattle, Washington, USA, July 8–12, pp. 485–492. ACM (2006)
12. Matsumoto, M., Nishimura, T.: Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modeling and Computer Simulation* 8(1), 3–30 (1999)
13. Cárdenas-Montes, M., Vega-Rodríguez, M.A., Gómez-Iglesias, A.: Real-world problem for checking the sensitiveness of evolutionary algorithms to the choice of the random number generator. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) *HAIS 2012, Part III. LNCS*, vol. 7208, pp. 385–396. Springer, Heidelberg (2012)
14. Charbonneau, P.: Genetic algorithms in astronomy and astrophysics. *The Astrophysical Journal Supplement Series* 101, 309–334 (1995)
15. Cárdenas-Montes, M., Mollá, M., Vega-Rodríguez, M.A., Rodríguez-Vázquez, J.J., Gómez-Iglesias, A.: Adjustment of observational data to specific functional forms using a particle swarm algorithm and differential evolution: Rotational curves of a spiral galaxy as case study. In: Sarro, L.M., Eyer, L., O’Mullane, W., De Ridder, J. (eds.) *Astrostatistics and Data Mining. Springer Series in Astrostatistics*, vol. 2, pp. 81–88. Springer, New York (2012)
16. García, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms’ behaviour: a case study on the cec’2005 special session on real parameter optimization. *J. Heuristics* 15(6), 617–644 (2009)
17. García, S., Fernández, A., Luengo, J., Herrera, F.: A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. *Soft Comput.* 13(10), 959–977 (2009)

The Artificial Bee Colony Algorithm Applied to a Self-adaptive Grid Resources Selection Model

María Botón-Fernández¹,
Miguel Á. Vega-Rodríguez², and Francisco Prieto Castrillo¹

¹ Ceta-Ciemat, Dept. Science and Technology, Trujillo, Spain
{[maria.boton](mailto:maria.boton@ciemat.es),[francisco.prieto](mailto:francisco.prieto@ciemat.es)}@ciemat.es

² Univ. Extremadura, Dept. Technologies of Computers and Communications,
Cáceres, Spain
mavega@unex.es

Abstract. Swarm intelligence algorithms are used to simulate the behaviour of non-centralized and self-organizing systems, which could be natural or artificial. Grid computing environments are distributed systems comprised heterogeneous and geographically distributed resources. This computing paradigm presents problems related to resources management (discovery, monitoring and selection processes) which are caused by its dynamic and changing nature. These problems lead to a bad application performance due to the fact that resources availability and characteristics vary over time. In recent years, several approaches based on *adaptation* and defined from a system point of view have been proposed. The present contribution is focussed on enhancing the grid resources selection process by providing a self-adaptive ability to grid applications. A selection model based on the Artificial Bee Colony algorithm is described. In contrast to other alternatives, the model is defined from a user point of view (the model has not control on the internal grid components). Finally, the approach is tested in a real European grid infrastructure. The results show that both a reduction in execution time and an increase in the successfully completed tasks rate are achieved.

Keywords: Artificial Bee Colony, Optimization, Grid Computing, Self-adaptive Ability, Swarm Intelligence.

1 Introduction

Grid computing environments [1] are distributed systems which have been increasingly used by the scientific community in the last decade. This type of infrastructure is formed by heterogeneous resources, with different geographical locations. Despite the advantages of such infrastructure, there are several problems related to resources management and task scheduling, which are caused by the characteristics and nature of grid systems. Grid applications compete for using non-dedicated resources and, also, they have to deal with two heterogeneity levels in grid systems. On the one hand, resources in a grid environment are owned by different centres, each of them with different operative systems and

different administrative domains. On the other hand, resources can be grouped based on their functionality. In a particular group, resources have different characteristics due to the fact that they belong to different centres. All these facts imply a variation in both the performance and the availability of resources (unpredictable systems), worsening the applications performance. Therefore, applications need to know the infrastructure's status in real-time during their execution; this way they can face the environmental changes. Furthermore, certain grid processes such as resources discovery, resources monitoring and resources selection should be improved for getting an autonomous system.

Nowadays, grid community is focused on designing/developing adaptive solutions [3]-[9] for addressing all these requirements. These solutions propose new frameworks, scheduling techniques, notification policies, etc. for improving the infrastructure performance (they are defined from a system point of view, which implies changes in the infrastructure behaviour). However, none of the proposed solutions have been used as a standard across the multiple grid platforms, so the problem persists. The present approach is focused on enhancing the selection process by choosing the most efficient resources during applications execution. We propose an *Efficient Resources Selection (ERS)* model which provides a self-adaptive capability to grid applications, by determining the resources that best fit the applications requirements during the execution. The model is defined from the user point of view, that is to say, it does not control the grid components and it does not modify the infrastructure behaviour. The efficiency of a resource is calculated by considering its processing time and its successfully finished tasks rate (as described in Section 3). The model combines this mathematical formulation with the Artificial Bee Colony (*ABC*) algorithm [2] for selecting resources in an efficient way (see Section 4). We denote our approach as *Efficient Resources Selection Model based on the Artificial Bee Colony (ERS-ABC)*. During the definition phase two objectives were established: a reduction in the application execution time and an improvement in the successfully finished tasks rate. Finally, the model is tested in a real grid infrastructure belonging to the European Grid Infrastructure (*EGI*). During the evaluation phase, two scenarios were defined to determine if the objectives fixed were accomplished. The *ERS-ABC* model is compared with the standard selection technique in European grid infrastructures, so that, the used baseline (gLite) follows the scheduling rules and policies of current European grid infrastructures.

The rest of the paper is structured as follows. Section 2 includes the related work. Section 3 describes the assumptions and mathematical formulation of the model. In Section 4 we specify how the *ABC* algorithm is used within the *ERS* model. The evaluation phase is exposed in Section 5. Section 6 concludes the paper.

2 Related Work

Today, the *adaptation* concept has become a widely used alternative for maintaining a suitable performance in grid computing environments. However, several

circumstances such as the different characteristics of grid components, the infrastructure heterogeneity and the changeable availability of resources, cause that applying *adaptation* in grid systems becomes a challenge itself.

There are some investigations based on the *adaptation* concept for solving resources management problems by improving the discovery, monitoring and selection processes. In [3] is described the *AppLeS (Application Level Scheduling Methodology)* project, which provides an adaptive ability to grid systems. A methodology for developing and deploying distributed high-performance applications in an adaptive form (an adaptive application scheduling) is proposed.

The work in [4] describes a framework for adaptive executions in Grid. It is based on *Globus*¹ and it is designed for handling grid jobs in an efficient way during execution. They use new scheduling techniques for maintaining a suitable performance level and for adapting to the changing conditions. Other work, exposed in [5], is focussed on improving the monitoring and discovery grid processes. The study introduces an approach for avoiding the *Information System (IS)*² from overloading. Besides, two adaptive notifications algorithms are exposed: a *sink-based algorithm* and a *utilization-based algorithm*. Both are based on *IS* availability and on data accuracy requirements.

In [6] it is presented an autonomous grid system which is adjusted dynamically to the application parallelism. Two rescheduling policies, suspension and migration, are described. A tolerance threshold is applied for determining which policy to use. The main idea in [7] is to gather information about resources processing times and resources communication during the application execution. This information is used to determine which resources injure the application performance. These resources will be replaced by the approach. The study in [8] presents a new adaptive data management architecture for *ARC (Advance Resource Connector)* Grid middleware which avoids bottleneck fails. The architecture is characterized by a three-layer structure that allows to use in a more efficient way the available bandwidth. Finally, in [9] a report of existing adaptive systems solutions is provided. The article includes suggestions for enabling autonomic operations in grid systems.

The works above discussed have a common characteristic: improving the grid infrastructures performance. Moreover, all these techniques (scheduling, rescheduling, notification, migration, etc.) have been designed from a system point of view (controlling grid components, modifying the infrastructure behaviour or architecture, using scheduling techniques or designing new policies as shown in Table 1). However, we propose an efficient and self-adaptive model for selecting the most efficient resources without modifying their behaviour or controlling them (i.e. the model does not use scheduling techniques neither allocation/migration policies). The model is defined from the user point of view, considering users limitations and applying their command set; it guides applications facing the environmental changes without modifying the infrastructure (see Table 1). Finally, it is expected to improve the infrastructure throughput.

¹ <http://www.globus.org/>

² The *Information System* registers useful information about grid resources.

Table 1. Main differences between current state of the art and the proposed *ERS-ABC* model. Please, observe that *ERS-ABC* does not have any control on the grid infrastructure.

Solution	Scheduling	New Policies	Change Infrs.	Control Components
AppLeS [3]	X	-	X	X
Framework [4]	X	-	X	X
M & D [5]	-	X	X	X
A Sys [6]	X	-	X	X
Living App [7]	-	-	X	X
ARC [8]	-	-	X	-
ERS-ABC	-	-	-	-

3 The Efficient Resources Selection Model

As stated, we propose a *ERS* Model which allows applications to self-adapt to grid changing conditions. The model is formed by both a mathematical formulation for obtaining the efficiency of grid resources and an intelligent selection process based on the *ABC* algorithm. During applications execution the model handles two work spaces: a task space J , which is constituted by n independent and parallel tasks, and a heterogeneous resource space R , which includes the available resources of the corresponding infrastructure. In particular, we measure the efficiency of a grid scheduler denoted as *Computing Element (CE)*. This component interacts with the compute nodes, determining in which one tasks are executed. Grid principles allow users to specify which *CE* will manage their tasks, so that, we can monitor them from a user point of view.

At the beginning of application execution, the model launches into execution a subset of J denoted as T . This way, we expect to promote a faster model learning. Then, a subset of R (known as RT) is selected for performing this task set. Resources in RT are chosen in a random way due to the fact that at that moment there are not efficiency metrics. Next, the model monitors the corresponding tasks. When a tasks t_α ends its execution, the efficiency of the corresponding resource r_α is measured. Then, by applying the *ABC* algorithm an efficient resource is chosen for a new task. All these steps are repeated until the whole space J is processed. Notice that every task has associated a *lifetime* lt ; this way, the model does not wait indefinitely for overloaded resources.

Concerning the efficiency value of a particular resource i two parameters are considered: on the one hand the historical value ϵ_i of successfully finished tasks³. On the other hand, the historical value μ_i of processing time used to perform these tasks. The historical value ϵ_i depends on the amount of successfully finished tasks SFt_i and on the total number of assigned tasks At_i as shown in Eq.1.

$$\epsilon_i = SFt_i / At_i . \tag{1}$$

³ Every task with a *Done* or *Aborted* status is considered as a finished task. Also tasks whose *lifetime* is over are included.

The processing time $T_{\{i,j\}}$ (Eq. 2.) of resource i is based on both the communication time $Tcomm_{\{i,j\}}$ between the resource i and other grid services during the execution of task j and on the computation time $Tcomp_{\{i,j\}}$ consumed for performing that task.

$$T_{\{i,j\}} = Tcomm_{\{i,j\}} + Tcomp_{\{i,j\}} . \quad (2)$$

All these $T_{\{i,j\}}$ values are used for attaining the processing time average value $\bar{\chi}_i$ for the CE (Eq.3). The parameter SFt_i is used in this equation for specifying the number of successfully finished tasks at that moment.

$$\bar{\chi}_i = \left(\sum_{j=1}^{SFt_i} T_{\{i,j\}} \right) / SFt_i . \quad (3)$$

Then, the historical value μ_i is calculated based on this $\bar{\chi}_i$ and on the *lifetime* lt fixed for performing tasks (see Eq.4).

$$\mu_i = (lt - \bar{\chi}_i) / lt . \quad (4)$$

Finally, the efficiency value E_i is measured by using both historical values ϵ_i and μ_i along with two relevance parameters a and b (as shown in Eq.5). These relevance parameters are introduced in the model for allowing users to specify the priorities conditions of their experiments.

$$E_i = (a \cdot \epsilon_i + b \cdot \mu_i) / (a + b) . \quad (5)$$

4 Applying the ABC Algorithm in the ERS Model

The *Artificial Bee Colony* is an evolutionary algorithm introduced by Dervis Karaboga [2]. In this algorithm the bee colony is composed of three types of bees: employed, onlooker and scout. After exploiting different food sources (nectar sources), employed bees return to the hive and dance to communicate the quality of the sources. Then, employed and onlooker bees choose (exploit) known food sources depending on the colony experience. On the other hand, scout bees explore new food sources, in this way, they choose the food sources in a random way without considering experience. In the *ERS-ABC* model we established the following assumptions:

- Our bees look for the most efficient resources, which are those resources with an efficiency value close to 1 (food sources with high nectar amount).
- A solution in *ERS-ABC* is an efficient resource.
- The employed bees exploit the q most efficient resources. The model groups these efficient resources in an employed set.
- For each employed resource the model estimates the probability of being exploited again. This probability value is based on resource's efficiency.

- The onlooker bees depend on employed bees information (experience). The model also handles a resource set for this type of bees (onlooker set).
- A scout bee uses a resource chosen in a random way.

Next, we specify the set of rules that govern the different sets of bees. When a resource r_α performs a task, the model determines which type of bee has exploited it. If it belongs to the employed bees the model evaluates the quality of the resource (i.e. how efficient it is). If the efficiency value exceeds the worst food source (worst efficiency value) memorized by the employed set the resource remains in the set. Otherwise, the model looks for an efficient resource which is not part of the employed set and discards r_α (mutation).

Concerning the onlooker set, at the beginning of the execution it is a replica of the employed set. During application execution it is updated whenever there is a change in the employed set. The corresponding upgrade method is based on employed resources' probability as shown in Figure 1. Also the well-known fitness proportionate selection is considered in this method. For obtaining the new onlooker set a mutation process is applied: an onlooker resource is replaced by its nearest and more efficient neighbour in space R . Notice that space R is sorted from highest to lowest efficiency value during application execution, so that, most efficient neighbours of a particular resource r_α will be located on its left.

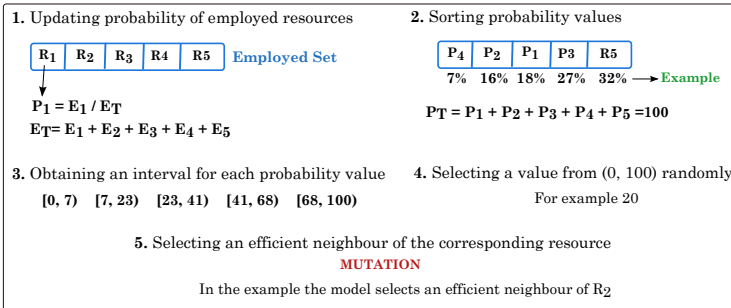


Fig. 1. Upgrade method for the onlooker set. It is based on employed resources probability. Steps 4 and 5 are repeated for every onlooker resource.

Regarding scout bees, every time a w percentage of space J is processed a scout bee is considered. This bee selects a resource from space R in a random way. Thus, all resources handled by these bees compose a candidate set of solutions. The new solution is chosen by applying a round robin technique over the candidate set.

Finally, in Figure 2 the execution flow of *ERS-ABC* is described. In the initialization phase (steps from 1 to 4) the two spaces are composed and the set T is launched into execution. Next, the model monitors the corresponding tasks. When a task t_α ends its execution, the efficiency value of the associated resource

PSEUDOCODE: ERS-ABC ALGORITHM

Input: application tasks, infrastructure resources
Output: set of solutions

1. Determine spaces J and R ;
 2. Prepare set T ;
 3. Select a set RT for T randomly;
 4. Launch T into execution;
 5. **while** there are unprocessed tasks **do**
 - 5.1. Monitor tasks;
 - 5.2. **If** a task ends its execution **then**
 - 5.2.1. Update resource efficiency value;
 - 5.2.2. Apply ABC selection process;
 - 5.2.3. Launch a new task;
 6. **End while**
-

Fig. 2. Pseudocode of *ERS-ABC* where the main steps of such model are summarized.

r_α is updated. Then, the *ABC* selection process is applied and an efficient resource is selected for performing a new task. These steps are repeated until all tasks $\in J$ are processed.

5 Performance Evaluation

As stated in Section 1, experiments are performed on a real European grid infrastructure, the *ES-NGI (National Grid Initiative of Spain)*⁴. In particular, we are affiliated to the Ibergrid project⁵ which a reasonable quantity of *CEs* for performing the model.

Two scenarios have been defined for determining if the main goals are achieved: a reduction in the total execution time and an increment of successfully finished tasks rate. In both scenarios, the *ERS-ABC* is compared with the standard selection technique (*TRS*) in grid systems. This selection is based on proximity and availability criteria. The method that performs that selection or tasks allocation is known in grid terminology as *match-making*.

Scenario 1

In this first scenario we want to determine the influence of size T within the model learning. For that reason, we fixed 5 tests with the next characteristics: the size of J is fixed at 200 tasks while the size of T varies in every test from 5 to 40 tasks (5, 10, 13, 20, 40). Experiments are executed 10 times for both versions (*ERS-ABC* and *TRS*). Then, each graphical point is the average value of these experiments. For larger size of T a faster model learning is expected.

The results (see Figure 3) show that *ERS-ABC* gets a better execution time with respect to *TRS*. In *ERS-ABC*, as we increase the size of T the total execution time is reduced. That means, the size of T influences the model learning

⁴ <http://www.es-ngi.es/>

⁵ <http://www.ibergrid.eu/>

making it faster. When we send bees to obtain information for a greater number of food sources (resources), in last tests, they reach a deeper knowledge of the infrastructure in shorter time. It must also be highlighted that the total execution time of *ERS-ABC* includes not only the application execution time but also the time spent by the model for monitoring grid resources and measuring their efficiency. However, *TRS* has an opposite behaviour. In first tests (sizes of 5 and 10) it gets its minimum values. Then, the execution time starts to increase progressively. In *TRS* resources are selected based on proximity and availability criteria, so that, it is assumed that in last tests there are few available resources (most of them are being used) and some of them are even overloaded.

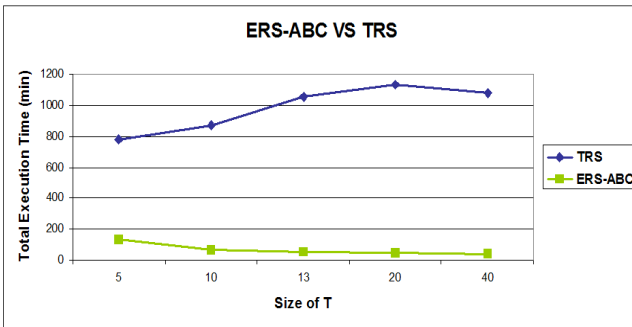


Fig. 3. Results obtained in the first scenario. The *ERS-ABC* gets better results than *TRS*.

Finally, in Table 2 some descriptive statistics concerning the total execution time for our *ERS-ABC* model are included. Notice that the coefficient of variation (*c.v.*) never surpasses 50% of the mean. Also, please, remember that grid infrastructures are very dynamic and changing, which motivates the values for the standard deviation. As stated, the model classifies better in last tests while in the first ones the resulting execution time depends much more on the first selection of resources.

Table 2. Statistical values of the *ERS-ABC* tests

Param.	Size 5	Size 10	Size 13	Size 20	Size 40
Mean	136.6	70.2	55.6	50.4	40.2
Standard Deviation	48.9	28.7	14.9	6.5	8.4
Coefficient Variation	36%	41%	27%	13%	21%

Scenario 2

The objective of this second scenario is to determine the range of applications in which is beneficial to apply our model. In grid computing, applications usually are composed of a high number of tasks. For that reason, we vary the size of *J*

from 50 to 500 tasks (50, 100, 200, 300, 400, 500). In this case, there are 6 tests in which both alternatives (*ERS-ABC* and *TRS*) are performed 10 times like in the previous scenario. The *ERS-ABC* again achieves an execution time reduction with respect to *TRS* (see Figure 4). The model also gets a better successfully finished tasks rate, especially in last tests in which *ERS-ABC* successfully performs almost all tasks. By contrast, *TRS* obtains worse execution time values as we increase J . We also include the corresponding statistical values (Table 3).

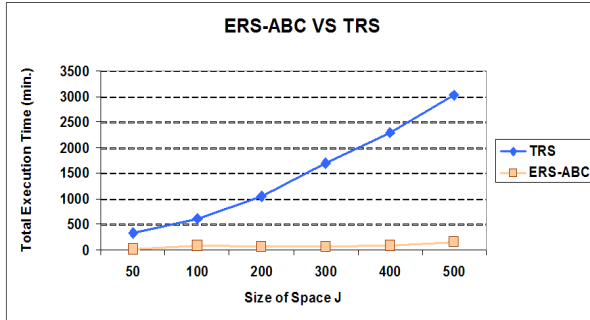


Fig. 4. Second scenario of the evaluation phase where *ERS-ABC* is compared with *TRS*

Table 3. Statistical values of the *ERS-ABC* tests in scenario 2

Param.	Size 50	Size 100	Size 200	Size 300	Size 400	Size 500
Mean	21.4	65.9	70.2	140.7	175.2	207.3
Standard Deviation	4.03	17.6	28.7	34.2	20.2	45.3
Coefficient Variation	19%	27%	41%	26%	12%	22%

Finally, we can conclude that the *ERS-ABC* model achieves the objectives pursued. For that reason, we consider that this is a feasible solution for grid applications.

6 Conclusions

The present contribution proposes an efficient resources selection model for enhancing the selection process in grid computing environments. The *ERS* model has been designed from the user point of view and is based on the *Artificial Bee Colony*. This model provides a self-adaptive capability, allowing applications to deal with the changing conditions. The evaluation phase and the resulting data show that our approach is a feasible solution for grid users, because it increases the successfully finished tasks rate as well as it reduces the applications execution time. Future work involves enhancing the proposed model by applying new algorithms and by considering/improving other grid services. Also we aim to compare our approach with other adaptive solutions.

Acknowledgement. María Botón-Fernández is supported by the PhD research grant of the Spanish Ministry of Economy and Competitiveness at the Research Centre for Energy, Environment and Technology (CIEMAT). The authors would also like to acknowledge the support of the European Funds for Regional Development.

References

1. Foster, I.: What is the Grid? A three Point Checklist. *GRIDtoday* 1(6), 22–25 (2002)
2. Karaboga, D.: An Idea based on Honey Bee Swarm for Numerical Optimization. Technical Report-tr06, Erciyes University, Turkey (2005)
3. Berman, F., Wolski, R., Casanova, H., Cirne, W., Dail, H., Faerman, M., Figueira, S., Hayes, J., Obertelli, G., Schopf, J., Shao, G., Smallen, S., Spring, N., Su, A., Zagorodnov, D.: Adaptive Computing on the Grid Using AppLeS. *IEEE Transactions on Parallel and Distributed Systems* 14(4), 369–382 (2003)
4. Huedo, E., Montero, R.S., Llorente, I.M.: A Framework for Adaptive Execution in Grids. *Software-Practice & Experience* 34(7), 631–651 (2004)
5. Keung, H.N.L.C., Dyson, J.R.D., Jarvis, S.A., Nudd, G.R.: Self- Adaptive and Self-Optimising Resource Monitoring for Dynamic Grid Environments. In: *Proceedings of the 15th International Workshop on Database and Expert Systems Applications, DEXA 2004, Washington DC, USA*, pp. 689–693 (2004)
6. Vadhiyar, S.S., Dongarra, J.J.: Self Adaptivity in Grid Computing. *Concurrency and Computation: Practice and Experience* 17(2-4), 235–257 (2005)
7. Groen, D., Harfst, S., Portegies Zwart, S.: On the Origin of Grid Species: The Living Application. In: Allen, G., Nabrzyski, J., Seidel, E., van Albada, G.D., Dongarra, J., Sliot, P.M.A. (eds.) *ICCS 2009, Part I. LNCS*, vol. 5544, pp. 205–212. Springer, Heidelberg (2009)
8. Cameron, D., Gholami, A., Karpenko, D., Konstantinov, A.: Adaptive Data Management in the ARC Grid Middleware. *Journal of Physics: Conference Series* 331 (2011)
9. Batista, D.M., Da Fonseca, L.S.: A Survey of Self-adaptive Grids. *IEEE Communications Magazine* 48(7), 94–100 (2010)

A Hybrid Algorithm Combining an Evolutionary Algorithm and a Simulated Annealing Algorithm to Solve a Collaborative Learning Team Building Problem

Virginia Yannibelli^{1,2} and Analía Amandi^{1,2}

¹ ISISTAN Research Institute, UNCPBA University

Campus Universitario, Paraje Arroyo Seco, Tandil (7000), Argentina

² CONICET, National Council of Scientific and Technological Research, Argentina

{vyannibe, amandi}@exa.unicen.edu.ar

Abstract. In this paper, we address a collaborative learning team building problem that considers a grouping criterion successfully analyzed in the context of software engineering courses. This criterion is based on taking into account the team roles of the students and on building well-balanced teams according to the team roles of their members. To solve the problem, we propose a hybrid algorithm. This algorithm incorporates a simulated annealing algorithm into the framework of an evolutionary algorithm with the aim of improving the performance of the evolutionary search. The simulated annealing algorithm adapts its behavior according to the evolutionary search state. The performance of the hybrid algorithm on ten data sets is compared with those of the algorithms previously proposed in the literature for solving the addressed problem. The obtained results show that the hybrid algorithm significantly outperforms the previous algorithms.

Keywords: collaborative learning, learning team building, team roles, hybrid algorithms, simulated annealing algorithms, evolutionary algorithms.

1 Introduction

In university software engineering courses, professors usually divide students into collaborative learning teams to develop collaborative learning tasks. These tasks are meant to supplement and enrich individual learning, and require the students to work together to solve a given problem (e.g., complete software projects) [2, 1]. In this context, the grouping criterion (i.e., the criterion to build learning teams) is important since the way in which a team is made up affects the learning level and the social behavior of the students belonging to the team as well as the performance of the team [2, 1]. Besides, the way in which the grouping criterion is applied (i.e., either manually or automatically) is important since many known grouping criteria require a considerable amount of knowledge, time and effort to be manually applied [7]. In these cases, it is possible to considerably reduce the workload of professors and optimize the collaborative learning team building through automation.

In the literature, different works have addressed the problem of building collaborative learning teams automatically [3, 4, 5, 6, 7, 8, 9]. These works differ in relation to the grouping criterion considered. Generally, these criteria take into account factors related to the learning state of the students, their learning style, thinking style and personality. However, to the best of our knowledge, only in few works [8, 9], the authors have considered a grouping criterion successfully evaluated in the context of software engineering courses.

In [8, 9], the authors address the problem of building collaborative learning teams automatically. As part of the problem, the authors consider a grouping criterion successfully analyzed in the context of software engineering courses: the grouping criterion recommended by the Belbin's team role model [10, 11]. This criterion is based on two central aspects. First, the criterion considers the team roles of the students. A team role is the way in which a person tends to behave, contribute and interrelate with others throughout a collaborative task. Second, the criterion is based on building well-balanced teams according to the team roles of their members. In this respect, the Belbin's model [10, 11] defines nine team roles and balance conditions. Different works that used the Belbin's model to study teams of students tasking software engineering group projects showed that considering the Belbin's roles [10, 11] can positively impact on the performance of the teams and on the learning level and the social behavior of the students [14, 12], and can provide a prediction of the performance of the teams based on the composition of the roles within the teams [13].

The collaborative learning team building problem addressed in [8, 9] is an NP-Hard optimization problem. For this reason, as reported in [8, 9], exhaustive search methods only can solve small instances of the problem in a reasonable period of time. Thus, heuristic algorithms have been proposed in the literature to solve the problem: an evolutionary algorithm was proposed in [8], and a memetic algorithm was proposed in [9] that incorporates a hill-climbing algorithm into the framework of an evolutionary algorithm. The memetic algorithm is the best of both algorithms [9].

In this paper, we address the learning team building problem described in [8, 9] with the aim of proposing a better heuristic algorithm to solve it. In this respect, we propose a hybrid evolutionary algorithm. This algorithm integrates a simulated annealing algorithm within the framework of an evolutionary algorithm. The behavior of the simulated annealing algorithm is self-adaptive based on observations from the state of the evolutionary search. The integration of a self-adaptive simulated annealing algorithm is meant to improve the performance of the evolutionary search [15, 17], and specifically, serves two purposes. In the early stages of the evolutionary search, when this search is diverse, the simulated annealing algorithm behaves like an exploitation process to fine-tune the evolutionary search. In later stages of the evolutionary search, when this search starts to converge, the simulated annealing algorithm behaves like an exploration process to diversify the evolutionary search.

We propose the above-mentioned hybrid evolutionary algorithm based on the following reasons. The hybridization of evolutionary algorithms with other search and optimization techniques has been proven to be more effective than the classical evolutionary algorithms in the resolution of a wide variety of NP-Hard problems [15, 17, 16, 19] and, in particular, in the resolution of learning team building problems [9].

Besides, the hybridization of evolutionary algorithms with simulated annealing algorithms has been shown to be more effective than the hybridization of evolutionary algorithms with hill-climbing algorithms in the resolution of different NP-Hard problems [15, 17]. Thus, we consider that the proposed hybrid evolutionary algorithm could outperform the heuristic algorithms previously proposed to solve the problem.

The remainder of the paper is organized as follows. In Section 2, we describe the addressed problem. In Section 3, we present the hybrid evolutionary algorithm proposed to solve the problem. In Section 4, we present the computational experiments carried out to evaluate the performance of the hybrid evolutionary algorithm and an analysis of the results obtained. In Section 5, we present related works. Finally, in Section 6 we present the conclusions of the present work.

2 Problem Description

In this paper, we address the collaborative learning team building problem presented in [8, 9]. This problem is described below.

A class S is made up of n students, $S = \{s_1, s_2, \dots, s_n\}$. The professor must divide the n students into g teams, $G = \{G_1, G_2, \dots, G_g\}$. Each G_i team is made up of a z_i number of member students, and each student can only belong to one team. As regards team size, students must be divided in such a way that the g teams have a similar number of students each. Specifically, the difference between the size of a team and the size of the other teams must not exceed one. The values of the terms S , n and g are known.

Regarding the students, it is considered that they naturally assume or play different team roles when taking part in a collaborative task. A team role is the way in which a person tends to behave, contribute and interrelate with others throughout a collaborative task. With respect of the team roles that can be played by the students, the nine team roles defined in Belbin's model [10, 11] are considered. Table 1 shows the nine roles and a brief description of the features of each.

According to Belbin's model [10, 11], it is considered that each student naturally play one or several of the nine roles described in Table 1. In this respect, the roles naturally played by each student are known data. These roles are obtained through the Belbin Team-Role Self-Perception Inventory (BTRSPI) developed by Belbin [11]. The BTRSPI determines the team roles of the persons by giving them self-evaluation tests [11].

As part of the problem, teams must be made up in such a way that the balance among the team roles of their members is maximized. This grouping criterion requires analyzing the balance level in the formed teams. To analyze the balance level, the balance conditions established by Belbin are considered [10, 11]. In relation to these conditions, Belbin [10, 11] states that a team is balanced if each role specified in his model is played naturally by at least one team member. In other words, in a balanced team, all team roles are naturally played. Further, Belbin states that each role should be naturally played by only one team member [10]. Belbin states that a team is unbalanced if some roles are not played naturally or if several of its members play the same role naturally (i.e., duplicate role) [10, 11].

Table 1. Belbin’s role characteristics

Role	Characteristics
Plant (PL)	Creative, imaginative, unorthodox. Solves difficult problems.
Resource Investigator (RI)	Extrovert, enthusiastic, communicative. Explore opportunities. Develops contacts.
Co-ordinator (CO)	Mature, confident, a good chairperson. Clarifies goals, promotes decision-making, delegates well.
Shaper (SH)	Challenging, dynamic, thrives on pressure. Has the drive and courage to overcome obstacles.
Monitor Evaluator (ME)	Sober, strategic and discerning. Sees all options. Judges accurately.
Teamworker (TW)	Co-operative, mild, perceptive and diplomatic. Listens, builds, averts friction.
Implementer (IM)	Disciplined, reliable, conservative and efficient. Turns ideas into practical actions.
Completer/Finisher (CF)	Painstaking, conscientious, anxious. Searches out errors and omissions. Polishes and perfects.
Specialist (SP)	Single-minded, self-starting, dedicated. Provides knowledge and skills in key areas.

The grouping criterion considered as part of the problem is modeled by Formulas (1), (2) and (3).

Formulas (1) and (2) formally express the balance conditions established by Belbin [10, 11]. Formula (1) analyzes the way in which a given r role is played within a given G_i team and gives a score accordingly. If r is naturally played by only one member of G_i team, then 1 point is awarded to G_i . Conversely, if r is not naturally played by any member of G_i , or otherwise r is naturally played by several members of G_i , then 2 points and p points are taken off respectively.

Formula (2) sets the balance level in a given G_i team. This balance level is established based on the scores obtained by G_i , through Formula (1), in relation to the nine roles. In this way, the greater the number of non-duplicate roles (i.e., roles played naturally by only one member of G_i), the greater the balance level assigned to G_i . Conversely, the fewer the number of roles played naturally, or the more duplicate roles, the lower the balance level assigned to G_i . The balance conditions established by Belbin [10, 11] can be seen in Formula (2). Using this formula, a perfectly balanced team (i.e., a team in which each of the nine roles is played naturally by only one team member) will obtain a level equal to 9.

Formula (3) maximizes the average balance level of g teams defined from the n students in the class. In other words, the objective of this formula is to find a solution (i.e., set of g teams) that maximizes the average balance level of g teams. This is the optimal solution to the problem addressed. In Formula (3), set C contains all the sets of g teams that may be defined from the n students in the class. The term G represents a set of g teams belonging to C . The term $b(G)$ represents the average balance level of the g teams belonging to set G . Then, Formula (3) uses Formula (2) to establish the balance level of each G_i team belonging to the G set. Note that in the case of a G set of perfectly balanced g teams, the value of the term $b(G)$ is equal to 9.

For a more detailed discussion of Formulas (1), (2) and (3), we refer to [8].

$$nr(G_i, r) = \begin{cases} 1 & \text{if } r \text{ is naturally played by only one member of } G_i \\ -2 & \text{if } r \text{ is not naturally played in } G_i \\ -p & \text{if } r \text{ is naturally played by } p \text{ members of } G_i \end{cases} \quad (1)$$

$$nb(G_i) = \sum_{r=1}^9 nr(G_i, r) \quad (2)$$

$$\max_{\forall G \in C} \left(b(G) = \frac{\sum_{i=1}^g nb(G_i)}{g} \right) \quad (3)$$

3 Hybrid Evolutionary Algorithm

To solve the addressed problem, we propose a hybrid evolutionary algorithm. This algorithm incorporates a simulated annealing stage into the framework of an evolutionary algorithm. The behavior of the simulated annealing stage is self-adaptive based on the diversity within the underlying evolutionary algorithm population. The incorporation of a self-adaptive simulated annealing stage pursues two aims. When the evolutionary algorithm population is diverse, the simulated annealing stage behaves like an exploitation process to fine-tune the solutions in the population. When the evolutionary algorithm population starts to converge, the simulated annealing stage changes its behavior from exploitation to exploration in order to introduce diversity in the population and thus to prevent the premature convergence of the evolutionary search. Thus, the evolutionary search is augmented by the addition of one stage of self-adaptive simulated annealing [15, 17].

The general behavior of the hybrid evolutionary algorithm is shown in Fig. 1 and is described as follows. Given a class of n students who shall be divided into g teams, the algorithm starts the evolution from a random initial population of feasible solutions. Each of these solutions codifies a feasible set of g teams which may be defined when the n students are divided. Then, each solution of the population is decoded (i.e., the set of g teams inherent to the solution is built), and evaluated according to the optimization objective of the problem by a fitness function. As explained in Section 2, the objective here is to maximize the balance level of the g teams formed from n students. Therefore, considering a given solution, the fitness function evaluates the balance level of the g teams represented by the solution. To perform that evaluation, the function is based on knowledge of the students' roles.

After the solutions of the current population are evaluated, a simulated annealing algorithm is applied to these solutions. The simulated annealing algorithm behaves like either an exploitation process or an exploration process depending on the diversity of the current population. Thus, the simulated annealing algorithm modifies the solutions of the current population. Then, a parent selection process is used to

determine which solutions of the population will compose the mating pool. The solutions with the greatest fitness values will have more chances of being selected. Once the mating pool is composed, the solutions in the mating pool are paired, and a crossover process is applied to each pair of solutions with a probability P_c to generate new feasible ones. Then, a mutation process is applied to each solution generated by the crossover process, with a probability P_m . The mutation process is applied with the aim of introducing diversity in the new solutions generated by the crossover process. Finally, a survival selection strategy is used to determine which solutions from the solutions in the population and the solutions generated from the mating pool will compose the new population.

This process is repeated until a predetermined number of iterations is reached.

```

BEGIN
  CREATE initial population;
  EVALUATE each candidate solution;
  REPEAT UNTIL ( number of iterations is reached ) DO
    IMPROVE solutions via Simulated Annealing;
    SELECT parents;
    RECOMBINE pairs of parents to produce offspring;
    MUTATE offspring;
    EVALUATE offspring;
    CREATE new population;
  OD
END

```

Fig. 1. General behavior of the hybrid evolutionary algorithm

3.1 Encoding of Solutions and Fitness Function

In relation to the encoding of solutions, we used a representation proposed in [7, 8, 9]. Each solution in the population is encoded as a list with a length equal to n . Each position j ($j = 1, \dots, n$) on this list contains a different student (i.e., repeated students are not admitted). Moreover, each student s_k ($k = 1, \dots, n$) may be in any position on the list. In short, the list is a permutation of the n students.

In order to decode the G set of g teams from the list, we used the decoding process proposed in [8, 9]. This process divides the list into g segments, considering that the difference between the size of a segment and the size of the rest of the segments must not exceed one. Each segment represents to a different team.

To evaluate a given encoded solution, a fitness function is used. This function decodes the G set of g teams represented by the solution. The decoding is carried out by applying the process above-described. Then, the function calculates the value of the term $b(G)$ (Formulas (3), (2), and (1)). This value represents the average balance level of the g teams composing the G set, and thus, determines the fitness level of the encoded solution.

3.2 Simulated Annealing Algorithm

In each iteration of the hybrid evolutionary algorithm, a simulated annealing algorithm is applied to each solution of the current population except to the best solution of this population. The best solution of the current population is maintained into this population. The simulated annealing algorithm applied here is an adaptation of the one proposed in [18], and is described as follows.

The simulated annealing algorithm is an iterative process that starts considering a given encoded solution s for the problem and a given initial value T_0 for a parameter called temperature. In each iteration, a new solution s' is generated from the current solution s by a move operator. If the new solution s' is better than the current solution s (i.e., the fitness value of s' is higher than the fitness value of s), the current solution s is replaced by s' . Otherwise, if the new solution s' is worse than the current solution s , the current solution s is replaced by s' with a probability equal to $\exp(-\text{delta} / T)$, where T is the current temperature value and delta is the difference between the fitness value of s and the fitness value of s' . Thus, the probability of accepting a new solution that is worse than the current solution mainly depends on the temperature value. If the temperature is high, the acceptance probability is also high, and vice versa. The temperature value is decreased by a cooling factor at the end of each iteration. The described process is repeated until a predetermined number of iterations is reached.

Before applying the simulated annealing algorithm to the solutions of the current population, the initial temperature T_0 is defined. In this case, the initial temperature T_0 is inversely proportional to the diversity of the current population, and this diversity is represented by the spread of fitnesses within the current population. Specifically, T_0 is calculated as detailed in Formula (4), where f_{max} is the maximal fitness into the current population and f_{min} is the minimal fitness into the current population. Therefore, when the population is very diverse, the value of T_0 is very low, and thus, the simulated annealing algorithm only accepts movements that improve the solutions to which it is applied, behaving like an exploitation process. When the population converges, the value of T_0 rises, and thus, the simulated annealing algorithm increases the probability of accepting worsening movements. A consequence of this is that the simulated annealing algorithm will try to move away from the solutions to which it is applied, exploring the search space. Eventually, the diversity of the population will increase, and thus, the temperature T_0 will decrease. Based on the above-mentioned, the self-adaptation of the simulated annealing algorithm to either an exploitation or exploration behavior is governed by the diversity of the population.

$$T_0 = \frac{1}{|f_{max} - f_{min}|} \quad (4)$$

In relation to the move operator used by the simulated annealing to generate a new solution from the current solution, we considered a feasible move operator for permutations of n elements. Specifically, we applied a move operator called insert mutation [15]. A detailed description of this operator can be found in [15].

3.3 Parent Selection, Crossover, Mutation and Survival Selection

To develop the parent selection, we used the tournament selection process [15] with a tournament size equal to five. This process is one of the most applied in the literature [15]. For a detailed discussion about this process and its advantages in relation to other parent selection processes, we refer to [15].

To develop the crossover and the mutation, we considered feasible processes for permutations of n elements. Specifically, we applied a crossover process called order crossover, with a probability of P_c , and a mutation process called swap mutation, with a probability of P_m . These processes are two of the most applied for permutations in the literature [15]. A detailed description of these processes can be found in [15].

To develop the survival selection, we applied the well-known steady-state process [15] with a replacement percentage of 80%. This process preserves the best solutions found by the hybrid evolutionary algorithm. For a detailed description of the steady-state process, we refer to [15].

4 Computational Experiments

To develop the experiments, we used the ten data sets presented in [8]. The main characteristics of each data set are shown in Table 2. Each data set contains a list of n students, and states a g number of teams to be built from the n students. Moreover, each data set defines team roles for each of its n students. These roles belong to the Belbin's model [10, 11] containing 9 roles as described in Table 1. Specifically, in each data set, each student of the data set has one or two roles. For a detailed description of the roles of each student in each data set, we refer to [8].

Each data set has a known optimal solution with a fitness level equal to 9. Note that an optimal solution with a fitness level equal to 9 defines a set of perfectly balanced g teams, according to the balance conditions established by Belbin [10, 11]. The known optimal solutions of the data sets are considered here as references.

The hybrid evolutionary algorithm was run 30 times on each of the data sets. After each one of the 30 runs, the algorithm provided the best solution of the last population. To perform these runs, the algorithm parameters were set with the values shown in Table 3. The parameters were fixed thanks to preliminary experiments that showed that these values led to the best and most stable results.

Table 4 presents the results obtained by the algorithm on each data set. The first column provides the name of each data set; the second column indicates the average fitness value of the achieved solutions for each data set; and the third column shows the average computation time of the runs performed on each data set. The experiments have been performed on a personal computer Intel Core 2 Duo at 3.00 GHz and 3 GB RAM under Windows XP Professional Version 2002. The algorithm has been implemented in Java programming language.

Table 2. Description of each data set

Data set	Number of participating students (n)	Number of teams (g)
1	18	3
2	24	4
3	60	10
4	120	20
5	360	60
6	600	100
7	1200	200
8	1800	300
9	2400	400
10	3000	500

Table 3. Parameter values of the hybrid evolutionary algorithm

Parameter	Value
Population size	80
Number of generations	200
<i>Simulated annealing algorithm</i>	
Number of iterations	25
Cooling factor	0.9
<i>Crossover process</i>	
Crossover probability P_c	0.8
<i>Mutation process</i>	
Mutation Probability P_m	0.15

Table 4. Results obtained by the hybrid evolutionary algorithm

Data set	fitness value	time (seconds)
1	9	0.29
2	9	0.721
3	9	5.81
4	9	9.24
5	9	21.46
6	8.97	29.27
7	8.86	103.43
8	8.77	190.1
9	8.74	301.69
10	8.7	405.118

We analyzed the results presented in Table 4 considering that each of the data sets has at least one optimal solution with a fitness level equal to 9. For each of the first five data sets, the algorithm has achieved an optimal solution in each of the runs. For each of the last five data sets, the algorithm has achieved an average fitness value that is higher than or equal to 8.7. This means that, for the last five data sets, the algorithm has achieved near-optimal solutions. We analyzed the composition of the obtained

solutions for the last five data sets. Based on this analysis, it is possible to say that each of these solutions contains a very high percentage of perfectly balanced teams.

In relation to the average computation time required by the algorithm, we may mention the following points. For the first six data sets, the average time required by the algorithm was lower than 30 seconds. For the last four data sets, the average time required by the algorithm was higher than 100 seconds and lower than 406 seconds. Taking into account the complexity of the problems inherent in the data sets, in particular the complexity of the problems inherent in the last four data sets, the average computation times required by the algorithm on the data sets are considered acceptable.

Based on these results, it is considered that the algorithm has achieved high-quality solutions in an acceptable period of time for each of the data sets.

4.1 Comparison with a Competing Algorithm

To the best of our knowledge, three algorithms have been previously proposed for solving the addressed problem: an exhaustive search method [8], a classical evolutionary algorithm [8], and a classical memetic algorithm [9]. According to the experiments reported in [8], the exhaustive search method and the evolutionary algorithm have been evaluated on the 10 data sets presented in Table 2 and have obtained the results that are shown in Table 5. In [9], the memetic algorithm only has been evaluated on the first 8 data sets presented in Table 2. The results obtained by the memetic algorithm for the first 8 data sets are presented in Table 5, as reported in [9]. In relation to the results of the memetic algorithm for the data sets 9 and 10, these results have been provided by the authors upon request and are presented in Table 5.

The experiments corresponding to the three above-mentioned algorithms have been carried out on a personal computer Intel Core 2 Duo at 3.00 GHz and 3 GB RAM under Windows XP Professional Version 2002. The three algorithms have been implemented in Java programming language.

Table 5. Results obtained by the algorithms previously proposed for the addressed problem

Data set	Exhaustive method		Evolutionary algorithm		Memetic algorithm	
	fitness value	time (s)	fitness value	time (s)	fitness value	time (s)
1	9	59.46	9	0.5537	9	0.42
2	9	189.27	9	1.3741	9	1.03
3	9	1072.59	9	11.0669	9	8.30
4	N/A	N/A	9	17.5976	9	13.20
5	N/A	N/A	8.8	40.8722	8.92	30.65
6	N/A	N/A	8.76	55.7548	8.86	41.82
7	N/A	N/A	8.7	196.9964	8.78	147.75
8	N/A	N/A	8.64	362.0328	8.68	271.52
9	N/A	N/A	8.61	574.6589	8.65	430.994
10	N/A	N/A	8.592	771.6553	8.62	578.74

Based on the results in Table 5, the memetic algorithm is better than the other two algorithms in relation to the average fitness value and the average computation time. Thus, the memetic algorithm is considered the best algorithm previously proposed in the literature for solving the addressed problem. Below, we compare the performance of the memetic algorithm with that of the hybrid evolutionary algorithm.

The results in Table 4 and Table 5 indicate that the hybrid evolutionary algorithm and the memetic algorithm have obtained the same average fitness value (i.e., an optimal fitness value) on the first four data sets (i.e., the less complex sets). However, the average fitness value obtained by the hybrid evolutionary algorithm on the last six data sets (i.e., the more complex sets) is higher than that obtained by the memetic algorithm. Besides, the average computation time required by the hybrid evolutionary algorithm for each data set is lower than that required by the memetic algorithm. We compare the average computation times required by the algorithms since the computer used here to evaluate the hybrid evolutionary algorithm has the same characteristics than the computer used in [9] to evaluate the memetic algorithm and the computer used in [8], and moreover, the algorithms have been implemented in the same programming language (i.e., Java).

The results above-analyzed suggest that the performance of the hybrid evolutionary algorithm on the 10 data sets is better than that of the memetic algorithm, in relation to the average fitness value and the average computation time. In order to ascertain the statistical significance of the differences observed, we have used a nonparametric statistical test called Friedman test [20]. First, the Friedman test was applied considering the average fitness values obtained by the hybrid evolutionary algorithm and the memetic algorithm on the 10 data sets. The ranking computed for the algorithms highlights the hybrid evolutionary algorithm as the best performing algorithm of the comparison (i.e., the rank of the hybrid evolutionary algorithm is 1.2 and the rank of the memetic algorithm is 1.8). The p -value computed through the Friedman statistic F (i.e., p -value is 0.05777957 and F is 3.6) suggests the existence of significant differences among the average fitness values obtained by the mentioned algorithms. Based on the results of the test, the hybrid evolutionary algorithm is better than the memetic algorithm in relation to the average fitness value, with a significance level $\alpha = 0.10$.

Then, the Friedman test was applied considering the average computation times required by the hybrid evolutionary algorithm and the memetic algorithm for the 10 data sets. The ranking computed for the algorithms highlights the hybrid evolutionary algorithm as the best performing algorithm of the comparison (i.e., the rank of the hybrid evolutionary algorithm is 1 and the rank of the memetic algorithm is 2). The p -value computed through the Friedman statistic F (i.e., p -value is 0.0015654 and F is 10) suggests the existence of significant differences among the average computation times required by the mentioned algorithms. Based on the results of the test, the hybrid evolutionary algorithm is better than the memetic algorithm in relation to the average computation time, with a significance level $\alpha = 0.01$.

Therefore, the performance of the hybrid evolutionary algorithm on the 10 data sets is better than that of the memetic algorithm, in relation to the average fitness value and the average computation time. The main reason for this is that the hybrid

evolutionary algorithm has a significant advantage over the memetic algorithm. This advantage is described below.

The hybrid evolutionary algorithm incorporates a simulated annealing algorithm into the framework of an evolutionary algorithm. The behavior of this simulated annealing algorithm is adaptive to either an exploitation or exploration behavior depending on the diversity within the evolutionary algorithm population. When this population starts to converge, the simulated annealing algorithm changes its behavior from exploitation to exploration, introducing diversity in the population and thus preventing the premature convergence of the evolutionary search. In contrast with the hybrid evolutionary algorithm, the memetic algorithm incorporates a hill-climbing algorithm into the framework of an evolutionary algorithm. This hill-climbing algorithm behaves like an exploitation process in each evolutionary cycle (i.e., fine-tunes the solutions obtained in each evolutionary cycle), even when the evolutionary algorithm population starts to converge. A consequence of this is that the hill-climbing algorithm accelerates the loss of diversity within the population, and therefore, usually leads to a premature convergence of the evolutionary search.

Based on the above-mentioned, the simulated annealing algorithm in the hybrid evolutionary algorithm prevents the premature convergence of the evolutionary search, whereas the hill-climbing algorithm in the memetic algorithm usually leads to a premature convergence of the evolutionary search. Thus, the hybrid evolutionary algorithm can reach better solutions than the memetic algorithm on the more complex data sets.

In relation to the temporal complexities of the hybrid evolutionary algorithm and the memetic algorithm, we may mention the following. The temporal complexity of the hybrid evolutionary algorithm is $O(mpsn)$, where m is the number of generations developed by the algorithm, p is the number of solutions in the population of the algorithm, s is the number of solutions generated by the simulated annealing algorithm from a given encoded solution, and n is the length of the encoded solutions.

The temporal complexity of the memetic algorithm is $O(mpln)$, where m , p , n have the meaning above-mentioned, and l is the number of solutions generated by the hill-climbing algorithm from a given encoded solution.

The difference between the temporal complexity of the hybrid evolutionary algorithm and that of the memetic algorithm is the following. The value of s in the hybrid evolutionary algorithm is much lower than the value of l in the memetic algorithm. Thus, the temporal complexity of the hybrid evolutionary algorithm is lower than that of the memetic algorithm.

5 Conclusions

We proposed a hybrid algorithm to solve the collaborative learning team building problem described in [8, 9]. This algorithm incorporates a simulated annealing algorithm into the framework of an evolutionary algorithm with the aim of improving the performance of the evolutionary search. The behavior of the simulated annealing algorithm is self-adapted to either an exploitation or exploration behavior depending

on the evolutionary search state. The computational experiments show that the hybrid algorithm outperforms the algorithms previously proposed for solving the addressed problem. In future works, we will evaluate other processes to develop the selection, crossover and mutation. Besides, we will investigate the incorporation of other search and optimization techniques into the framework of the evolutionary algorithm.

References

1. Barkley, E.F., Cross, K.P., Howell Major, C.: Collaborative learning techniques. John Wiley & Sons, Inc. (2005)
2. Michaelsen, L.K., Knight, A.B., Fink, L.D.: Team-based learning: A transformative use of small groups in college teaching. Stylus Publishing, Sterling (2004)
3. Christodoulopoulos, C.E., Papanikolaou, K.A.: A Group Formation Tool in an E-Learning Context. In: 19th IEEE ICTAI 2007, pp. 117–123. IEEE Press, New York (2007)
4. Wang, D.Y., Lin, S.S.J., Sun, C.T.: DIANA: A computer-supported heterogeneous grouping system for teachers to conduct successful small learning groups. *Computers in Human Behaviors* 23(4), 1997–2010 (2007)
5. Cavanaugh, R., Ellis, M., Layton, R., Ardis, M.: Automating the Process of Assigning Students to Cooperative-Learning Teams. In: 2004 American Society for Engineering Education Annual Conference & Exposition. American Society for Engineering Education, Salt Lake City (2004)
6. Graf, S., Bekele, R.: Forming Heterogeneous Groups for Intelligent Collaborative Learning Systems with Ant Colony Optimization. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 217–226. Springer, Heidelberg (2006)
7. Lin, Y.T., Huang, Y.M., Cheng, S.C.: An automatic group composition system for composing collaborative learning groups using enhanced particle swarm optimization. *Computers & Education* 55(4), 1483–1493 (2010)
8. Yannibelli, V., Amandi, A.: A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context. *Expert Systems with Applications* 39(10), 8584–8592 (2012)
9. Yannibelli, V., Amandi, A.: A memetic algorithm for collaborative learning team formation in the context of software engineering courses. In: Cipolla-Ficarra, F., Veltman, K., Verber, D., Cipolla-Ficarra, M., Kammüller, F. (eds.) ADNTIIC 2011. LNCS, vol. 7547, pp. 92–103. Springer, Heidelberg (2012)
10. Belbin, R.M.: *Management Teams: Why They Succeed or Fail*. Butterworth-Heinemann, Oxford (1981)
11. Belbin, R.M.: *Team Roles at Work*. Butterworth-Heinemann, Oxford (1993)
12. Winter, M.: *Developing a group model for student software engineering teams*. Master's thesis. University of Saskatchewan (2004)
13. Johansen, T.: *Predicting a Team's Behaviour by Using Belbin's Team Role Self Perception Inventory*. PhD thesis. University of Stirling (2003)
14. Stevens, K.: *The Effects of Roles and Personality Characteristics on Software Development Team Effectiveness*. PhD thesis. Faculty of Virginia Polytechnic Institute and State University (1998)
15. Eiben, A.E., Smith, J.E.: *Introduction to Evolutionary Computing*, 2nd edn. Springer (2007)
16. Talbi, E.-G. (ed.): *Hybrid Metaheuristics*. SCI, vol. 434. Springer, Heidelberg (2013)

17. Rodríguez, F.J., García-Martínez, C., Lozano, M.: Hybrid Metaheuristics Based on Evolutionary Algorithms and Simulated Annealing: Taxonomy, Comparison, and Synergy Test. *IEEE Transactions on Evolutionary Computation* 16(6), 787–800 (2012)
18. Yannibelli, V., Amandi, A.: Hybridizing a multi-objective simulated annealing algorithm with a multi-objective evolutionary algorithm to solve a multi-objective project scheduling problem. *Expert Systems with Applications* 40(7), 2421–2434 (2013)
19. Blum, C., Puchinger, J., Raidl, G.R., Roli, A.: Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing* 11(6), 4135–4151 (2011)
20. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* 1(1), 3–18 (2011)

Addressing Constrained Sampling Optimization Problems Using Evolutionary Algorithms

Pilar Caamaño, Gervasio Varela, and Richard J. Duro

Integrated Group for Engineering Research, Universidade da Coruña, Spain
C/ Mendizabal s/n, Campus de Esteiro, Ferrol
{pcsobrino, gervasio.varela, richard}@udc.es

Abstract. In this work we address the solution of a particular category of problems, denoted as Constrained Sampling problems, using evolution. These problems have not usually been addressed using EAs. They are characterized by the fact that the fitness landscape evaluation is not always straightforward due to the computational cost or to physical constraints of the specific application. The decoding phase of these problems usually implies some type of physical migration from the constructs generated to obtain the fitness of the parents towards those required to obtain the fitness of the offspring. As a consequence, it is not instantaneous and requires a series of steps. Most traditional evolutionary algorithms ignore the information on the fitness landscape that can be obtained from these intermediate steps. We propose a series of modifications that can be applied to most EAs that allow improving their efficiency when applied to this type of problems. This approach has been tested using some common real-coded benchmark functions and its performance compared to that of a standard EA, specifically a Differential Evolution algorithm.

Keywords: Optimization, Evolutionary Algorithms, Constrained Sampling Problems, Constrained Sampling Evolutionary Algorithm.

1 Introduction

Evolutionary Algorithms (EAs) are metaheuristics that have been widely applied to search and optimization problems. Nevertheless, in spite of their great performance, their use has been discarded in applications where the fitness evaluation is not straightforward or where it is very expensive in terms of computational time.

To deal with the problem of costly evaluations, the authors of the field have proposed the use of surrogate models [9], which are approximations to the fitness functions using coarser models with lower computational requirements. Several techniques to approximate fitness such as approximation levels, approximate model management schemes or model construction techniques [5] have been analyzed for application to problems where the calculation of the fitness is extremely expensive, as in the case of, for example, structural design optimization [9], protein structure prediction [10] or protein design and drug design [11].

However, surrogate models are not appropriate in those problems where the bottleneck is not really the evaluation of the individuals but rather their decoding (usually taken as part of the evaluation phase). In the field of EAs the resolution of this type of problems has also been discarded, as, by definition, for the application of an EA it is assumed that every point of the search space can be directly, or almost directly, decoded, i.e., it is possible to create the instance of the solution represented by the individual easily or, at least in a reasonably easy way.

This is not always the case. There are applications that do not meet this requirement. As an example, let the objective function of our application be to find the point of the atmosphere with the highest pollution level. In this case, it is necessary to measure the pollution level at each point of the atmosphere (or at least at those the EA determines that should be sampled) in order to find the optimum one. In other words, to complete the task, we either have sensing units in every point of the atmosphere that needs to be sampled (which is not practical) or, if we have less sensing units than points to be sampled, we need to move the sensors around. The use of EAs is not very efficient in these situations as the points that should be sampled are usually far from the points over which the sampler is located after measuring the previous fitness value. In this detection example, the decoding phase involves moving the sampler to a physical position of the solution space. In other words, in order to obtain the fitness of an offspring its decoding is only possible through a series of transformations (in this case motions) from the decoded version of one of the parents or from that of another individual in the parent population.

This case exemplifies the strict definition of the concept of Constrained Sampling Problems (CS-Problems), that is, search or optimization applications in which it is necessary to physically reach a specific point of the solution space through a series of transformations from the decoded version of a previous point in order to be able to produce a fitness value so as to continue with the evolutionary cycles of an EA. Physically reaching a point in the solution space may involve, in some cases, moving a sensor to that point. In others it may imply constructing a physical entity. What is important here is that the process is not instantaneous and requires following a trajectory (either through motion or through a series of construction or transformation steps) from the decoded version of a previous individual that leads to the solution point. In these cases the process of following the trajectory, which is basically the decoding process of the individual, is the real bottleneck of the EA. Thus, in general, a CS-Problem could be defined as a problem in which the decoding of the individuals implies a bottleneck in the application of the EA. Target tracking, source detection or path planning in unknown environments are problems that often belong to this category.

EAs assume that reaching any point in the solution space is simple and fast and, therefore, there is usually no restriction on what point is chosen to be sampled next (regardless of where the sensing unit is). This makes EAs quite inefficient in these situations and, as a consequence, these problems have usually been addressed using other types of metaheuristics like Particle Swarm Optimization (PSO) [1, 2]. EAs have only been used indirectly in these cases by some authors that concentrated on the off-line evolution of control policies that were able to use information from the environment for real-time improvements [6, 8].

Other problems can be considered CS. Examples are the evolution of modular robots [4], in which, every time an individual has to be evaluated, a new robot needs to be constructed (usually from pieces that were used in a previous instance and that must be disassembled), or Facility Layout Problems (FLPs) [12], in which the individuals of the population represent the configuration of production plants and, in order to measure the fitness of each one, a simulation model has to be constructed. The use of traditional EAs in these problems implies ignoring information about the landscape that could be gathered during the decoding phase, as each step of this process produces a new solution. EAs do not usually measure the fitness of these intermediate individuals as they are not produced by the reproduction operators. The approach presented in this work proposes a set of modifications for most EAs that allow their efficient application to CS-Problems.

This paper is structured as follows; section 2 is devoted to the description of the solution proposed, Constrained Sampling EAs. In section 3, the results obtained after the validation of the new proposal are presented. Finally some conclusions and future work are discussed in section 4.

2 Constrained Sampling Evolutionary Algorithms

As indicated before, the constraints and requirements of the applications described in the previous section lead to inefficiencies in the direct application of the usual types of EAs. To address this issue, in this paper we propose a series of modifications and additions to the common execution cycle of a standard EA. The elements and relationships needed for the implementation of the new concept, the Constrained Sampling Evolutionary Algorithm (CS-EA), are explained in detail in the following sections.

2.1 CS – EA Components

The design of the concept of the CS-EA is based on the necessary modifications to be performed over a standard EA in order to deal with the constraints CS-Problems introduce. Thus, we start from any EA whose operators will be used to explore the fitness landscapes. In order to make the description of the CS-EA clearer, this element will be denoted as *basis-EA*.

The common execution cycle of any EA requires an offspring population ($P'(t)$, where t is the current generation) to be evaluated in order to be compared to the parent population ($P(t)$) before a replacement phase that leads to the production of the next generation parent population ($P(t+1)$). The definition of CS problems usually implies following lengthy trajectories (sequences of transformations) in solution space from the individuals in $P(t)$ to those in $P'(t)$ before the individuals belonging to $P'(t)$ can be evaluated. Thus, $P'(t)$ can be taken as the target of the trajectories and in what follows we will denote it as $T(t)$, that is, the *target population*. We propose using the information gathered by moving along these trajectories in solution space as new

information to be considered by the EA in order to make it more efficient. To this end, two additional populations of individuals will be considered by the CS-EA:

- A population that includes all the positions or points of the search space that have been evaluated up to that point, denoted as the *evaluated population* (hereafter, $E(t)$).
- A population of *evaluating entities*, these elements are the successive transformations of the individuals in $P(t)$ when moving towards $T(t)$ and they can be used to measure the fitness of the landscape points they go through and send the measurements to the *evaluated population*. Each *evaluating entity* (\vec{c}) has an associated individual from the *target population* (\vec{t}) that guides the transformation of the along a trajectory (basically, the *current individual* changes by moving towards the *target*). It must be highlighted that the number of *evaluating entities* is independent of the *basis-EA* population size.

In addition to these two populations, the CS-EA also implies changes with respect to regular EAs in the way the fitness landscape is covered. The following elements are necessary for this objective:

- A strategy to select the target individual that is assigned to each *evaluating entity*, hereafter the *target selection strategy* (τ), is required. In other words, it is necessary to somehow decide how we choose the target towards which a given *evaluating entity* moves.

$$\tau: \mathbb{R}^n \times \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^n, \tau(\vec{c}, T(t)) = \vec{t}', \vec{t}' \in T(t)$$

- A definition of the rules that define the modifications that can be performed over the parameters of the *current individual* of the *evaluating entities*. These rules are always adapted to the specific application and they have been called *application rules* (σ).

$$\sigma: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n, \sigma(\vec{c}, \vec{t}) = \vec{c}', |\vec{c} - \vec{t}| \leq |\vec{c}' - \vec{t}|$$

The only requirement in the definition of the *application rules* is the following:

$$\sigma^n(\vec{c}) = \vec{t}, n \in \mathbb{N}, n < \infty$$

2.2 CS – EAs Execution Cycle

The CS-EA is very similar in its operation to the most common EAs. Nevertheless, the two new populations must be considered so that EAs can be applied to CS-Problems with a reasonable computational cost.

To begin, each generation the CS-EA has to wait until the *evaluated population* reaches, at least, a minimum size of n individuals, corresponding to the population size used by the *basis-EA*. The parent population of the *basis-EA* is then filled with the first n individuals of the *evaluated population*, i.e. $P(t) = \{\vec{x}_i, i = 1 \dots n \wedge \vec{x}_i \in E(t)\}$. At this point the CS-EA runs a generation; the selection phase is executed to select the individuals from the parent population that will be reproduced. The reproduction operators are used to generate a new offspring population that will be stored

as the *target population*, that is, $T(t) = \{\vec{x}_i, i = 1 \dots m \wedge \vec{x}_i \in P'(t)\}$. After this the *evaluating entities* start to follow a trajectory from the current parent population to the target population, providing fitness values for the points along the way and sending them to the *evaluated population*. Once the *evaluated population* has, again, n *evaluated individuals*, the replacement operators compare the *parent population* to the *evaluated individuals* selected from the *evaluated population*, creating the new *basis-EA* population that leads to a new generation of the CS-EA, i.e., $P(t + 1) \subset E(t + 1) \cup P(t)$. The flowchart of Figure 1 represents the sequence of steps followed by the CS-EA each generation.

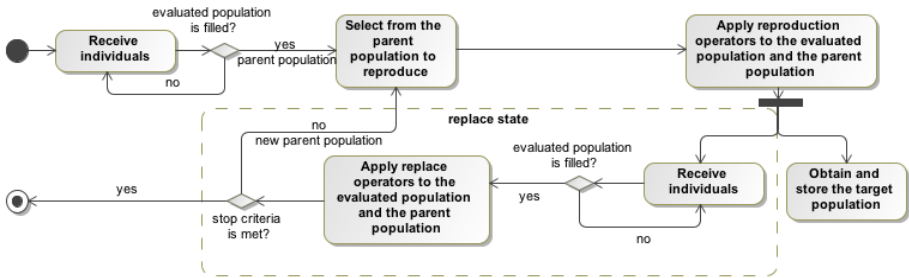


Fig. 1. Constrained Sampling EA flow chart

As previously mentioned, due to the computational cost or to physical constraints of the applications, the individuals of the *basis-EA* cannot directly measure the fitness of the solutions proposed by the *target population*. The *evaluating entities* go through the fitness landscape following a trajectory guided by the individuals of the *target population*. Each instant of time, the *current individuals* of the *evaluating entities* are modified according to the *application rules*. The *evaluating entities* measure the fitness of their new positions in search space and send them to the *evaluated population* (this fitness value is not that of the target, but that of a point closer to the target than the initial value). Finally, according to the *target selection strategy*, a new target individual is assigned to the *evaluating entity*. The behavior of the *evaluating entities* is represented in the flowchart of Figure 2.

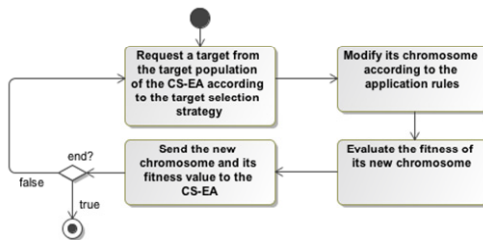


Fig. 2. Evaluating entities behavior flowchart

Summarizing, the *target population* generated using the *basis-EA* operators guides the changes in the *evaluating entities* that feed the *evaluated population*, which is used by the EA in the evolutionary process to generate new targets.

In addition to the possibility of using this approach in applications where a standard EA requires lengthy runtimes, the design of the concept of CS-EA described in this work presents the following advantages:

- There are no constraints on the way that the *evaluated population* is filled. The evaluation time of each *evaluating entity* could be different, thus, some of them could possibly generate more new evaluated individuals than others. This is possible because the behavior of the *evaluating entities* is decoupled from the execution cycle of the *basis-EA*.
- The operation of the CS-EA proposed in this work has been designed to allow the coupling of any EA. As this operation procedure may be used in many different applications with different features and EA requirements, the use of one or another depends on the needs of the final application.

3 Some Experiments

In this section we will compare the performance of a Constrained Sampling EA and that of a common EA over five well known benchmark optimization functions. We will first describe the experimental setup used to analyze the capabilities of the new approach as compared to a standard EA. After this description we will provide a discussion and analysis of the results obtained.

3.1 Experimental Setup

Test-Bed Configuration

Due to space limitations, only five benchmark functions will be presented in this paper. The selected functions are the *Sphere Model* ($f1$), and *Ackelys's* ($f2$), *Rastrigin's* ($f3$), *Rosenbrock's* ($f4$) and *Schwefel's* ($f5$) functions. The first one is unimodal, being the other four functions multimodal with different optima distributions. The analytical expressions and the characteristics of these functions may be found in [14].

The algorithms are compared in terms of efficiency, effectiveness and precision using the speedup, the absolute error and the genotypic error [3], respectively. To ensure that the results are statistically significant, 50 independent runs of each experimental setup configuration were run. Their stop criterion was based on the maximum number of function evaluations (FEs), as it is the one used in the most popular algorithm competitions, the maximum number of FEs is $n \cdot 10^5$, where n is the dimension of the problem.

Algorithm Configuration

In this case, a Differential Evolution algorithm (DE) [13] was used as the *basis-EA* as it has shown the best average performance in the most popular algorithm competitions of the field [7, 14]. To fairly compare and analyze the results, the DE with the same

configuration will be used as the standard EA. In [3], the authors analyzed the performance of DE in terms of different optimization functions including the five that will be used in this work. Thus, the configuration of the algorithms in this work is the same as the one used in [3]. For all the experiments the population size was set to 50 individuals and their dimension was set to 2.

To analyze the influence of the number of *evaluating entities* on the performance of the two algorithms, the tests were carried out using 5, 10 and 20 *evaluating entities* for the CS-EA. Again, to incorporate the constraints imposed by CS-problems, when evaluating the standard EA we also assumed a limitation in the number of evaluating entities. This means that the offspring population was evaluated, as in the case of any CS-problem, by using the number of samplers available (5, 10 or 20) until the whole offspring population had been covered. Our initial hypothesis is that the larger the number of evaluating entities, the less time required to complete the task. Also, by increasing the number of evaluating entities, the exploration capabilities of the algorithm increase, so the absolute and genotypic error should also decrease.

Regarding the *target selection strategy*, in this work two very different strategies were compared. The first one, from now on denoted as *random*, selects a random individual from the target population. The second one, hereafter denoted as *closest*, chooses the individual from the target population that is closest to the current one. The *target selection strategy* is also in charge of determining when the *target* is changed. In the case of the standard EA, the *target* is only changed when the *evaluating entities* reach it, as it is mandatory to measure its fitness to fill the *evaluated population*. This is not the case of the CS-EA as the *evaluating entities* fill the evaluated population as they move along the search space. Nonetheless, to fairly compare the algorithms, in this work the *target* of the CS-EA approach will only be modified when the *evaluating entities* reach it. Notice that this decision is unfair for the CS-EA approach because its exploration capabilities decrease when forced to reach the targets, thus reducing its performance with respect to when it changes targets freely.

To finish with the configuration of the algorithm, the *application rules* need to be defined. In these experiments, we simulate a target-finding problem where the target is the optimum of the benchmark function. Thus, the *evaluating entities* will be moved throughout the search space like physical samplers with the same physical constraints, i.e., their movements are limited by a maximum step length, here set to 0.01 units, which represents 0.35% of the search space.

3.2 Results

The results obtained after carrying out the experiments are presented in this section. All the runs were solved successfully. In all of them, except for some cases when considering Schwefel's function, the absolute error was below $1.0e-6$ ¹. In the case of Schwefel's function the absolute error (see Table 1) was higher when the random target selection strategy was chosen. However, by analyzing the results provided in

¹ More results provided in the experiments are available at

<https://github.com/GII/JEAF/wiki/Constrained%20Sampling%20EAs>.

terms of distance to the optimum (Genotypic error in Table 2) the difference between the two methods are not very significant.

The results represented in figures 3-5 correspond to the speedup comparisons between the two approaches. From top to bottom, the plots display the results when 5, 10 and 20 evaluating entities are used. As shown, the CS-EA approach is always superior to the standard EA in terms of efficiency, speeding up the search significantly. The difference between the two models decreases as the number of evaluating entities increases because of the convergence properties of the standard EA, which imply slightly shorter average trajectory lengths as the number of Evaluating Entities increases.

Table 1. Absolute error provided by the algorithms for Schwefel's function

f	Eval. entities	Random target selection strategy			Closest target selection strategy		
		EE5	EE10	EE20	EE5	EE10	EE20
Schwefel's	<i>St - EA</i>	1.4e-07 ± 0e+00	1.4e-07 ± 0e+00	1.4e-07 ± 0e+00	1.4e-07 ± 0e+00	1.4e-07 ± 0e+00	1.4e-07 ± 0e+00
	<i>Cs - EA</i>	3.9e+00 ± 2.0e+01	1.4e-02 ± 7.4e-02	4.4e-04 ± 1.5e-03	1.4e-07 ± 4.5e-14	1.4e-07 ± 8.5e-12	1.4e-07 ± 1.0e-13

Table 2. Genotypic error provided by the algorithms in the Schwefel's experiments

f	Eval. entities	Random target selection strategy			Closest target selection strategy		
		EE5	EE10	EE5	EE10	EE5	EE10
Schwefel's	<i>St - EA</i>	1.3e-07 ± 4.9e-10	1.3e-07 ± 5.1e-10	1.3e-07 ± 4.3e-10	1.3e-07 ± 6.5e-10	1.3e-07 ± 7.1e-10	1.3e-07 ± 5.8e-10
	<i>Cs - EA</i>	3.0e-02 ± 2.0e-01	1.6e-04 ± 6.6e-04	4.4e-04 ± 1.5e-03	1.3e-07 ± 4.7e-10	1.3e-07 ± 6.1e-09	1.3e-07 ± 4.9e-10

The experiments where the random selection strategy is used are the ones that show more significant differences. In the worst case, the Sphere function, the CS-EA is approximately 2.5 times faster than the St-EA. The best cases are the experiments where Schwefel's function is solved, in which the CS-EA is approximately 12 times faster than the standard approach. The differences are more remarkable in functions f2-f5, which are multimodal. This is due to the fact that they require a more exploratory behavior from the EA. In other words, the population needs to be more spread out throughout the solution space and thus the trajectories are lengthier than in the case of f1, which is unimodal. Among the multimodal functions, f2, f3 and f4 present a topographical structure (see [3] for more details) that requires less explorative capacities than function f5. This is the reason why the difference between the CS-EA and the St-EA is more remarkable in the case of function f5. In this case the CS-EA approach explores the solution space in a short span of time, as it takes advantage from the information it gathers during the trajectories while the St-EA does not. The efficiency results of the standard EA improve when using the closest target selection strategy, due to the fact that the trajectories covered by the evaluating entities are shorter than in the random target selection strategy experiments.

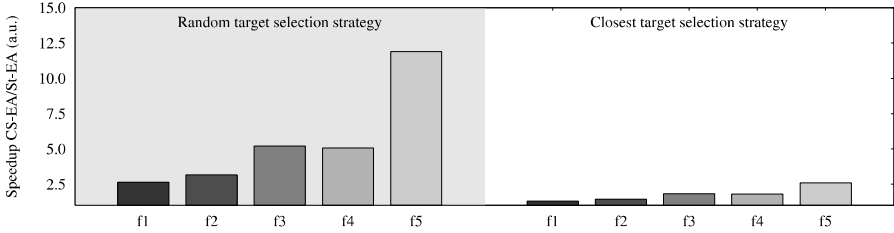


Fig. 3. Speedup comparison between the two algorithms when 5 evaluating entities are used

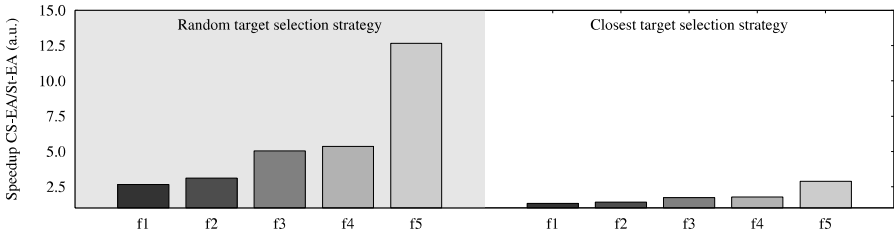


Fig. 4. Speedup comparison between the two algorithms when 10 evaluating entities are used

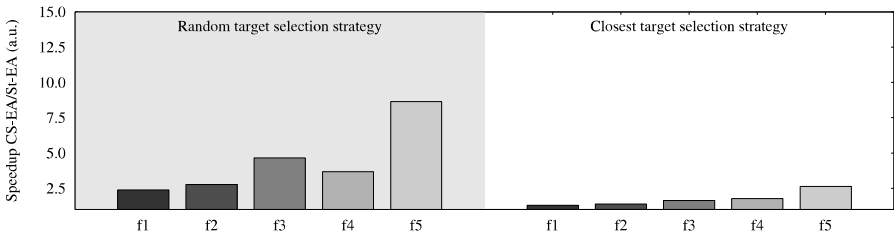


Fig. 5. Speedup comparison between the two algorithms when 20 evaluating entities are used

4 Conclusions and Future Work

In this work we have detected a niche of problems where the use of Evolutionary Algorithms was not very popular due to the poor efficiency of their application. We have denoted them as Constrained Sampling problems. With the aim of improving the performance of the EAs when solving this type of problems, several modifications in their operation have been proposed here leading to a new algorithm, the CS-EA. This algorithm can be used in applications where it is useful to take advantage of the steps carried out while the individuals are being decoded by using information from the landscape that could be gathered during this process. This information is not usually considered by the most common EAs.

The CS-EA model was tested using some common real-coded benchmark functions. Its performance was compared to the one provided by a standard EA, in this case a Differential Evolution algorithm. The results show better efficiency for the CS-EA as compared to the standard approach. Moreover, the CS-EA performance is not influenced by the number of *evaluating entities* or the type of function, which is not the case of the standard EA that requires more *evaluating entities* in order to be competitive in terms of speedup.

We are now working on the characterization of the CS-EA in terms of higher dimensionalities so as to analyze its scalability. In this sense, our first experiments, with 10 dimensions, show that the CS-EA is, approximately, 2000 times faster than the St-EA in terms of speedup. We are also working in the application of the CS-EA to real world problems like the ones exemplified in the introduction to test its capabilities in real constrained environments.

Acknowledgments. This work was partially funded by the Spanish MICINN through project TIN2011-28753-C02-01 and the Xunta de Galicia and European Regional Development Funds through projects 09DPI012166PR and 10DPI005CT.

References

1. Banks, A., Vincent, J., Phalp, K.: Particle Swarm Guidance System for Autonomous Unmanned Aerial Vehicles in an Air Defence Role. *Journal of Navigation* 61(01), 9–29 (2008)
2. Bao, Y., Fu, X., Gao, X.: Path planning for reconnaissance UAV based on particle swarm optimization. In: 2010 Second International Conference on Computational Intelligence and Natural Computing Proceedings, CINC, vol. 2, pp. 28–32. IEEE (September 2010)
3. Caamaño, P., Bellas, F., Bécerra, J.A., Duro, R.J.: Evolutionary algorithm characterization in real parameter optimization problems. *Applied Soft Computing* 13(4), 1902–1921 (2013)
4. Faiña, A., Orjales, F., Bellas, F., Duro, R.J.: First Steps towards a Heterogeneous Modular Robotic Architecture for Intelligent Industrial Operation. In: Workshop Reconfigurable Modular Robotics: Challenges of Mechatronic and Bio-Chemo-Hybrid Systems, IROS, p. 6 (2011)
5. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing* 9(1), 3–12 (2005)
6. Kuroki, Y., Young, G.S., Haupt, S.E.: UAV navigation by an expert system for contaminant mapping with a genetic algorithm. *Expert Systems with Applications* 37(6), 4687–4697 (2010)
7. Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello, C.C., Deb, K.: Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. *Journal of Applied Mechanics* 41 (2006)
8. Nikolos, I.K., Valavanis, K.P., Tsourveloudis, N.C., Kostaras, A.N.: Evolutionary algorithm based offline/online path planner for UAV navigation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 33(6), 898–912 (2003)
9. Ong, Y.S., Nair, P.B., Keane, A.J.: Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA Journal* 41(4), 687–696 (2003)

10. Piccolboni, A., Mauri, G.: Application of evolutionary algorithms to protein folding prediction. In: Hao, J.-K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D. (eds.) AE 1997. LNCS, vol. 1363, pp. 123–135. Springer, Heidelberg (1998)
11. Schneider, G.: Neural networks are useful tools for drug design. *Neural Networks* 13(1), 15–16 (2000)
12. Singh, S.P., Sharma, R.R.K.: A review of different approaches to the facility layout problems. *The International Journal of Advanced Manufacturing Technology* 30(5-6), 425–433 (2006)
13. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4), 341–359 (1997)
14. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *KanGAL Report 2005* (2005)

Genetic Algorithm-Based Allocation and Scheduling for Voltage and Frequency Scalable X MOS Chips

Zorana Banković¹ and Pedro López-García^{1,2}

¹ IMDEA Software Institute, Madrid, Spain

² Spanish Council for Scientific Research (CSIC), Spain
{zorana.bankovic, pedro.lopez}@imdea.org

Abstract. In this work we present a novel approach, based on genetic algorithms, for automatic scheduling and allocation of tasks in a multi-processor multi-threaded architecture, together with an assignment of the appropriate voltage and frequency of each processor in a way the overall energy consumption is optimized and all task deadlines are met. The approach deals with scheduling, allocation and voltage and frequency assignment at the same time, and provides good solutions in a very short time. As far as we know, this is the first approach that supports two levels of parallelism: multi-processor and multi-thread.

1 Introduction

Dynamic power consumption due to switching activity in digital CMOS circuits can be expressed with the following formula: $P = \alpha C_{eff} V^2 f$, where C_{eff} is the effective capacitance, V is the operating voltage, f is the operating frequency, and α the switching factor. If we can decrease the voltage supply and the operating frequency, the dynamic power will decrease significantly. On the other hand, static power, which is the result of the leakage currents, also decreases quadratically with voltage [7]. Thus, voltage decrease can achieve significant power and energy savings. This process is known as Dynamic Voltage and Frequency Scaling (DVFS). However, it slows down the operation of the circuit, and has to be applied in a way the required deadlines are still fulfilled. Furthermore, the process introduces additional latencies, so we have to develop a set of requirements that define the applicability of this approach.

The objective of this work is to optimize energy consumption through optimal scheduling and allocation for a set of tasks on X MOS chips, which are multiprocessor and multithreaded voltage and frequency scalable architecture. In X MOS chips threads are pipelined in a four-stage pipeline, where in each stage one instruction from different thread is executed, so in essence we can say that the threads also run in parallel. Thus, we deal here with two levels of parallelism. We assume that different processors can have different (V, f) setting, while the threads running on the same processor at the same time must have the same (V, f) setting.

Given a set of tasks and their corresponding deadlines, the objective is to provide a scheduling and allocation, and also assign voltage and frequency for each processor that would optimize the energy, while respecting the deadlines. The tasks are heterogeneous, and they in general have different starting time and deadline. We assume that there is no precedence between tasks, and no preemption. In order to solve the problem, we need to have safe estimates of power consumption of each task, as well as its execution time. Since this work falls into the ENTRA project [1], whose main task is to provide the programmer the estimation of the energy consumption of his/hers program at compile time, we assume that there exists an analysis that would give us this information, as necessary input. On the other hand, there is a great body of work about time analysis, so we assume that the analyzer will provide us this information as well.

The general problem of scheduling and allocation is NP-hard. In order to solve it, different heuristic algorithms have been developed since they are capable of obtaining sub-optimal solutions in real time. Many of them use genetic algorithms (GA) [3,4,9] due to its fast exploration of the search space, which allow quickly finding acceptable solutions. For this reason, our scheduler will also be based on GA. We will provide an appropriate solution representation that captures the two levels of parallelism, i.e. at both processor and thread level, and in the same run performs allocation and scheduling and identifies appropriate (V, f) setting in real time. As far as we know, this is the only solution for this type of problems.

The rest of the work is organized as follows. Section 2 details the sources of power consumption, while Section 3 explains the problem that is being solved and draws the constraints that are the basis for generating the solution. Section 4 details the implemented solution, while Section 5 explains the experimentation environment and presents the most significant results. Section 6 presents the most relevant related work, and finally, Section 7 draws the most important conclusions.

2 CPU Power Consumption

The energy required to complete a (set of) task(s) in time T on one processor, given the frequency f and voltage V is defined by:

$$E_{cpu,f,V} = \int_{t_0}^{t_0+T} P_{cpu,f,V}(t) dt \quad (1)$$

where $P_{cpu,f,V}$ is the time varying XCore power at (V, f) setting. This power can be calculated as:

$$P_{cpu,f,V}(t) = P_{cpu,V}^{fix} + P_{idle,f,V} + P_{cpu,f,V}^{act}(t) \quad (2)$$

where $P_{cpu,V}^{fix}$ is the portion of the power that includes PLL and leakage [7], which is the part that only depends on voltage, not on frequency. $P_{idle,f,V}$ is the power spent when the processor is not executing any application. For a certain fixed

(V, f) setting, the sum of these two does not change in time, so in the further text we will call it standing power consumption, $P_{cpu, V, f}^{std}$. This power can be easily obtained by measuring the CPU power when there are no running applications for each (V, f) setting. On the other hand, $P_{cpu, f, V}^{act}(t)$ is the active power spent on switching activity during the execution of the application(s). Finally, we can write:

$$P_{cpu, f, V}(t) = P_{cpu, f, V}^{std} + P_{cpu, f, V}^{act}(t) \quad (3)$$

which put in 1 gives the energy consumed during time T :

$$E_{cpu, f, V} = P_{cpu, f, V}^{std} T + \sum_{i=1}^M P_{i, f, V} T_i \quad (4)$$

where $P_{i, f, V}$ is the power spent by the application i , which is executed during time of T_i , and M is the number of threads, i.e. the maximal number of applications that can be executed on one processor at certain moment. In the cases when the threads can finish more than one application within time T , formula 4 would have the following form:

$$E_{cpu, f, V} = P_{cpu, f, V}^{std} T + \sum_{i=1}^M \sum_{j=1}^K P_{ij, f, V} T_{ij} \quad (5)$$

where K is the maximal number of applications a thread can execute in time T .

3 Problem Description

Problem Definition

Given a set of concrete tasks, provide optimal scheduling and (V, f) pair(s) for each processor in order to optimize energy consumption.

Input

- Set of tasks with their corresponding deadlines.
- Set of possible (V, f) pairs.
- Available hardware: n - number of processors, m - number of threads per processor.

Output

Viable scheduling and allocation that optimizes energy.

In the following text we assume the notation where variables are expressed using upper case letters, while constants are expressed using lower case letters.

3.1 Timing Constraint

In general, for each new frequency $F_{new, i}$ of each processor i , the following should remain valid:

$$\forall i \in [1, n], \forall j \in [1, m], T_{oh, i} + \frac{C_{ij}}{F_{new, i}} \leq D_{ij} \quad (6)$$

where T_{oh} is the time overhead introduced by DVFS and C_{ij} is the number of clock cycles needed to execute the application j on processor i , giving its execution time to be $\frac{C_{ij}}{F_{new,i}}$. This is reasonable to assume, given that in X MOS there are no pipeline stalls, nor cache misses, since there is no cache memory. We further have:

$$\forall i \in [1, n], T_{oh,i} = t_{oh_V} + T_{oh_f,i} \approx t_{oh_V} + \frac{10}{F_{old,i}} + \frac{2}{F_{new,i}} \quad (7)$$

where t_{oh_V} is the time overhead of performing voltage scaling (assumed to be constant), while T_{oh_f} is the time overhead of performing frequency scaling, which takes 10 clock cycles at most of the old clock, and two cycles of the new clock [6]. Finally, from 6 and 7 we get the timing constraints set:

$$\forall i \in [1, n], \forall j \in [1, m], F_{new,i} \cdot (C_{ij} + 2) \leq D_{ij} - t_{oh_V} - 10/F_{old,i} \quad (8)$$

where we consider that we know t_{oh_V} , and both F_{old} and F_{new} can take one value from the finite set of the pre-established values (V, f) .

3.2 Energy Minimization Constraint

The second set of requirements is derived from the condition of reducing the total energy during some known time t , high enough so that it permits the termination of all the applications. This implies the following condition:

$$\forall i \in [1, n], \forall j \in [1, m], t \geq \max_{i,j} D_{ij} \quad (9)$$

Thus, for each processor, we have:

$$\begin{aligned} \sum_{i=1}^n E_{old} &\geq \sum_{i=1}^n E_{new} \Rightarrow \\ \sum_{i=1}^n p_{i,cpu,F_{old,i},V_{old,i}}^{std} \cdot t + \sum_{i=1}^n \sum_{j=1}^m p_{ij,V_{old,i},F_{old,i}} \cdot \frac{C_{ij}}{F_{old,i}} &\geq \\ \sum_{i=1}^n e_{oh} + \sum_{i=1}^n p_{i,cpu,F_{new,i},V_{new,i}}^{std} \cdot t + \sum_{i=1}^n \sum_{j=1}^m p_{ij,V_{new,i},F_{new,i}} \cdot \frac{C_{ij}}{F_{new,i}} & \end{aligned} \quad (10)$$

where e_{oh} is the energy spent on voltage and frequency scaling, $p_{ij,V_{old,i},F_{old,i}}$ and $p_{ij,V_{new,i},F_{new,i}}$ are estimated total power consumptions of the application j on XCore i in the different (V, f) settings, while p_{cpu}^{std} is the standing power explained in Section 2 in different settings. Finally, from 10, we get:

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^m \left(\frac{p_{ij,V_{new,i},F_{new,i}}}{F_{new,i}} - \frac{p_{ij,V_{old,i},F_{old,i}}}{F_{old,i}} \right) \cdot C_{ij} &\leq \\ t \cdot \sum_{i=1}^n (p_{i,cpu,F_{old,i},V_{old,i}}^{std} - p_{i,cpu,F_{new,i},V_{new,i}}^{std}) - n \cdot e_{oh} & \end{aligned} \quad (11)$$

where the only unknown parameters are C_{ij} .

4 Proposed Solution

Our solution for optimal scheduling and allocation is based on GA. We have used steady-state GA, where the number of individuals of the population is the same in every generation and in every generation 60% of the population with lowest objective values is replaced with newly created individuals. Custom roulette wheel selector is used for the selection process. In the following text we will explain other important aspects of its implementation in more detail.

Individual. The starting point, and one of the most important parts, in designing a GA-based solution is always a representation of a solution, i.e. individual. In our case, the solution contains information about temporal and spatial allocation of each task. In other words, for each processor and each of its threads we should have an ordered (in time) set of tasks. However, since in this work we deal with DVFS, we have to add the information about the (V, f) state of each processor. All the threads on the same processor have the same (V, f) setting in the same moment, but we assume that different processors can have different (V, f) setting, in order to solve the most general problem.

We can look at a solution to the scheduling problem as a permutation of the task identifiers, where their order also stands for the order of their temporal execution, assuming that each task has a unique identifier. On the other hand, in order to solve the allocation problem, i.e. on which thread (and which processor) each task is executed, we can add delimiters to the permutation of the task IDs that would define where the tasks are being executed, i.e. processor, thread and (V, f) setting (the tasks between two delimiters are executed on the right-side one). In order to be able to distinguish delimiters from the task, they are used as negative three-digit numbers, where the first digit stands for the processor, the second for the thread on that processor, and the third for the processor (V, f) setting (there is a finite number of settings). Part of a solution is depicted in Fig. 1, where tasks with IDs 1, 2, 5 and 7 are executed in that order on the thread 4 of the core 2, with the (V, f) setting marked as 4. In the most general case, the order of delimiters is random. However, if two consecutive delimiters that belong to the same processor have different settings, this means that they are not being executed in parallel, since the state has to be changed. Representing a solution in the described way has provided us with a relatively simple solution, which will not introduce great overhead when executing GA.



Fig. 1. Solution Representation

Population Initialization. We have used a heuristic when initializing the population in order to provide some good quality individuals from the beginning. According to it, the task is added to the thread in the way the total resulting energy up to the moment is minimal. However, the total energy is calculated for the time equal to the farthest deadline for each thread. In this way, more weight is given to the static power overhead. Thus, the objective of this heuristic is to promote delaying the execution of each task towards its deadline through minimizing the energy overhead. However, since in general GAs benefit from great variety of solutions, we also introduce random solutions. During the initialization process, each individual randomly chooses between heuristic and a randomly generated solution, where the heuristic has slightly bigger possibility to be chosen (0.6).

Solution Perturbations. Given that all the tasks and all the delimiters are different, different solutions are always a permutation of a set of tasks and the set of delimiters. This gives us the opportunity to use some of the permutation-based crossover operators, and in this case we are using the partial match crossover, since it performed better in the terms of objective function than the cycle crossover, and slightly better than order crossover in the terms of objective function and execution time. Since the order of delimiters is not important in the most general case, this operator provides at the same time variety in consecutive changes of (V, f) settings, as well as moving tasks from one thread to another. Regarding mutation, it is implemented in the way that two random threads exchange two random tasks with a small probability.

Objective Function. Since the aim is to minimize total energy, the objective function is the total energy consumed for executing the given set of tasks and it is calculated as presented in Section 2. However, the execution time for each thread is taken as the farthest deadline of its tasks, in order to take full advantage of the DVFS possibility. Furthermore, we have to be sure that the solution is viable, i.e. that all given deadlines are met. We deal with this problem through the penalization of the inviable solutions by multiplying their energy by 10. In this way, viable solutions will always have higher objective function and thus higher probability of surviving to the next generation.

5 Experimental Evaluation

5.1 Testing Environment

XMOS Chips. The target architecture for this work are XMOS chips. However, the same approach can be followed for any kind of DVFS-enabled multi-processor architectures. In the case of XMOS chips, both voltage and frequency scaling are possible and both introduce time overhead. All of their chips provide the possibility

of frequency scaling due to the existence of a programable frequency divider. The time overhead introduced when changing the frequency is not more than 10 cycles of the old clock, plus two more cycles of the new clock.

On the other hand, only the XS1-SU01A-FB96 [8] chip provides the possibility of voltage scaling due to the existence of two DC-DC converters whose output voltage can belong to the range (0.6V, 1.3V). In order to apply DVFS, we should have a list of Voltage-Frequency (V, f) pairs or ranges that provide correct chip functioning. The latency in this case is not controllable, and can be estimated in the following way. Since the switching frequency of the converter is 1MHz, and assuming we have linear control, the bandwidth should be 1/10 to 1/7 of it, i.e. 150kHz in the best case. Thus, the time it takes for the output voltage to stabilize is 1/150kHz, which is around $6\mu s$.

We have experimentally concluded that the XMOS chips can function properly in six (V, f) settings given in the first two columns of Table 1. In order to include the possibility of shutdown, we could include the state (0(V),0(MHz)) and take the wake-up time as the latency of changing the state, and proceed in the same way. For the purpose of this experiment, we assume that we have four different processors, where each processor has eight different threads.

Task Set. For the purpose of this initial experiment we have used the tasks from the well known SPECCPU2006 [2] benchmark. The input dataset is composed of 200 tasks randomly chosen, where each is one from the benchmarks. Each task is independent. The same reasonable deadline is assigned to each task, so it provides the possibility of applying DVFS. Their execution time is measured on a general purpose computer, and the execution time on an XMOS chip is estimated to be $T_{measured} \cdot \frac{f_{XMOS}}{f_{gp}}$. This estimation is based on the assumption that the total number of execution cycles is the same in both cases, and that it is representative of the total execution time. While this is true for the XMOS chip, in the general purpose computer this is not the case due to cache misses, pipeline stalls, etc. Thus, in the future we would have to profile the tasks on the XMOS chips, or use static analysis. It is important to point out that the duration of the tasks, as well as their energy, are much bigger than both time and energy overhead of DVFS scaling, so in this experiment the overhead will not be a limiting factor.

Table 1. Typical power consumption for each processor state

$V(V)$	$f(MHz)$	$P_{dyn}(mW)$	$P_{st}(mW)$
0.95	500	117.325	18.05
0.87	400	78.7176	15.138
0.8	300	49.92	12.8
0.8	150	24.96	12.8
0.75	100	14.625	11.25
0.7	50	6.37	9.8

Since this work represents an initial study of the approach, we have taken that the power consumption of each task is the typical X MOS power consumption given in [7]. The estimations for different (V, f) settings are estimated by scaling with voltage and frequency in the case of dynamic power, while the static power is scaled with voltage, i.e. $P_{dyn} = \frac{P_{dyn}^{base} \cdot f_{new} \cdot V_{new}^2}{f_{base} \cdot V_{base}^2}$ and $P_{st} = \frac{P_{st}^{base} \cdot V_{new}^2}{V_{base}^2}$. These values are given in Table 1 for each (V, f) setting. However, it is assumed that in the future the analyzer will give us an estimation of power consumption of each task.

5.2 Obtained Results and Discussion

Our genetic algorithm has been executed on 500 individuals, during 100 generations. Greater number of individuals does not provide significantly better

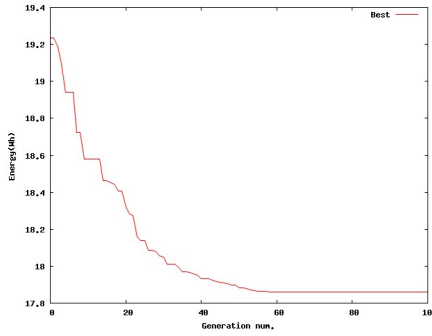


Fig. 2. Evolution of the best objective value

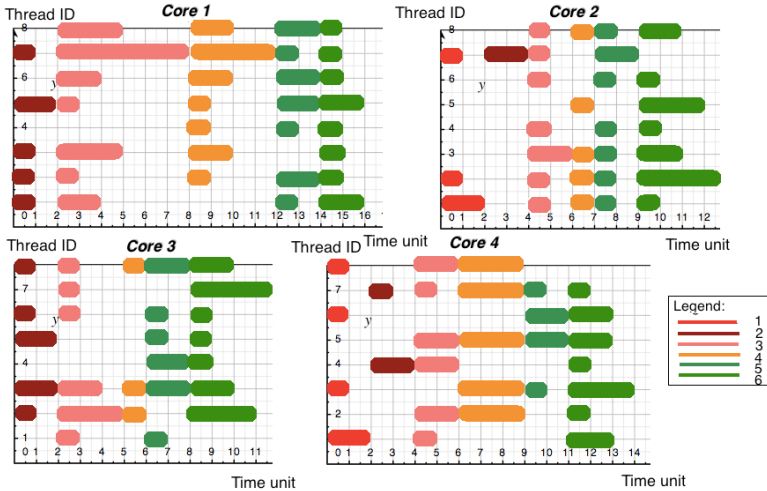


Fig. 3. A Scheduling and Allocation Solution per Core with Assigned DVFS setting

solution. In Fig. 2 we can see that the best objective value does not significantly change during the last iterations. The objective value is given in Wh. Under these settings, the total execution time of the algorithm is around six minutes on an Intel Dual Core machine, with 2.4GHz clock.

The average saving achieved in this way is 33.94%, with standard deviation of 0.56%, compared to the same scheduling and allocation without the DVFS. Speaking in terms of statistical significance, we can be 95% sure that the obtained savings will belong to the interval (33.02%, 34.86%). A typical solution is presented in Fig. 3. Separate (V, f) settings are distinguished with different colors, where the settings 1-6 correspond to the ones given in Table 1, and one time unit corresponds to one task. As we can observe, the majority of the tasks are executed in low power settings 4, 5 and 6.

6 Related Work

Since DVFS can provide significant energy savings, its optimal usage has been extensively studied. Some examples divide scheduling and allocation in two separate tasks, such as the one given in [11], where in the first step the allocation problem is solved using Linear Programming, while in the second the scheduling problem is solved for separate processors using Bin Packing. Another solution [3] solves the scheduling problem using GA, while it integrates DVFS in the fitness function. However, we believe that more optimal solutions could be achieved when solving scheduling and allocation at the same time, while also accounting for the DVFS. There is one example of GA-based scheduling [4] that combines scheduling, allocation and power management in one task. However, it deals only with voltage scaling.

There is also a significant group of publications on using GAs for optimal scheduling and allocation in multiprocessor systems with DVFS possibility. An example given in [9] treats the problem as bi-objective, where they want to minimize both energy and make span. The same objective is solved in another work [10], but using the island model of parallel GA populations. Yet, in this work our aim is to optimize the energy while meeting the deadlines, but our approach can easily be adapted to work as multi-objective. Another solution [5] treats the problem from two opposite points of view: in the first one, optimizes the energy given the scheduler length, while in the other optimizes the scheduling length given the energy bound. Finally, none of the solutions does not include the possibility of two levels of parallelism, where each processor can have a number of different threads executing in parallel.

7 Conclusions

In this work we have presented a solution for optimizing energy consumption for a multiprocessor and multithreaded architecture. The solution performs scheduling and allocation as one task, and deals with two levels of parallelism, which is the only solution of this kind as far as we know.

The solution will form part of the tool developed within the ENTRA project, when we will be able to include the power and time estimates provided by the ENTRA static analyzer. There are also possibilities to further improve the performances of the solution. For example, XMOS chips have the possibility to automatically reduce the frequency of the processor if all of its threads are waiting for an event, and in this way decrease the energy consumption even further. This feature will be included in future versions of our scheduler, as well as the possibility of shutting off separate components while they are not active.

Acknowledgements. The research leading to these results has been supported by the European FP7/2007-2013 318337 *ENTRA* project, and the Spanish TIN2012-39391-C04-01 *STRONGSOFT* project.

References

1. Entra project (2013), <http://entraproject.eu/>
2. Speccpu2006 (2013), <http://www.spec.org/cpu2006/>
3. Ying, C.-T., Yu, J.: Energy-aware genetic algorithms for task scheduling in cloud computing. In: 2012 Seventh ChinaGrid Annual Conference, ChinaGrid, pp. 43–48 (2012)
4. Kianzad, V., Bhattacharyya, S.S., Qu, G.: Casper: an integrated energy-driven approach for task graph scheduling on distributed embedded systems. In: 16th IEEE International Conference on Application-Specific Systems, Architecture Processors, ASAP 2005, pp. 191–197 (2005)
5. Kumar, P.R., Palani, S.: A dynamic voltage scaling with single power supply and varying speed factor for multiprocessor system using genetic algorithm. In: 2012 International Conference on Pattern Recognition, Informatics and Medical Engineering, PRIME, pp. 342–346 (2012)
6. XMos Ltd. Xs1-l active energy conservation (April 2010)
7. XMos Ltd. Estimating power consumption for xs1-l devices (May 2012)
8. XMos Ltd. Xs1-su01a-fb96 datasheet (November 2012)
9. Mezmaz, M., Lee, Y.C., Melab, N., Talbi, E., Zomaya, A.Y.: A bi-objective hybrid genetic algorithm to minimize energy consumption and makespan for precedence-constrained applications using dynamic voltage scaling. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2010)
10. Mezmaz, M.-S., Kessaci, Y., Lee, Y.C., Melab, N., Talbi, E.-G., Zomaya, A.Y., Tuyttens, D.: A parallel island-based hybrid genetic algorithm for precedence-constrained applications to minimize energy consumption and makespan. In: 2010 11th IEEE/ACM International Conference on Grid Computing, GRID, pp. 274–281 (2010)
11. Paterna, F., Acquaviva, A., Caprara, A., Papariello, F., Desoli, G., Benini, L.: An efficient on-line task allocation algorithm for QoS and energy efficiency in multicore multimedia platforms. In: Design, Automation Test in Europe Conference Exhibition, DATE, pp. 1–6 (March 2011)

Second Order Swarm Intelligence

Vitorino Ramos¹, David M.S. Rodrigues^{2,3}, and Jorge Louçã³

¹ *LaSEEB* – Evolutionary Systems and Biomedical Eng. Lab., ISR – Robotic and Systems Institute, Technical University of Lisbon (IST), Av. Rovisco País, 1 Torre Norte, 6.21, 1049-001 Lisbon, Portugal

vitorino.ramos@ist.utl.pt

² The Open University, Milton Keynes, United Kingdom

david.rodrigues@open.ac.uk

³ *The Observatory* - ISCTE-IUL, Lisbon University Institute (IUL), Av. Forças Armadas, 1649-026 Lisbon, Portugal

jorge.l@iscte.pt

Abstract. An artificial Ant Colony System (ACS) algorithm to solve general-purpose combinatorial Optimization Problems (COP) that extends previous AC models [21] by the inclusion of a negative pheromone, is here described. Several Traveling Salesman Problem (TSP) were used as benchmark. We show that by using two different sets of pheromones, a second-order coevolved compromise between positive and negative feedbacks achieves better results than single positive feedback systems. The algorithm was tested against known NP-complete combinatorial Optimization Problems, running on symmetrical TSPs. We show that the new algorithm compares favorably against these benchmarks, accordingly to recent biological findings by Robinson [26,27], and Grüter [28] where "No entry" signals and negative feedback allows a colony to quickly reallocate the majority of its foragers to superior food patches. This is the first time an extended ACS algorithm is implemented with these successful characteristics.

Keywords: Self-Organization, Stigmergy, Co-Evolution, Swarm Intelligence, Dynamic Optimization, Foraging, Cooperative Learning, Combinatorial Optimization problems, Symmetrical Traveling Salesman Problems (TSP).

1 Introduction

Research over hard NP-complete *Combinatorial Optimization Problems* (COP's) has, in recent years, been focused on several robust bio-inspired meta-heuristics, like those involving *Evolutionary Computation* (EC) algorithmic paradigms [1-3], as well as other kind of heuristics and approximation algorithms [4-5]. One particularly successful well-know meta-heuristic [6] approach is based on *Swarm Intelligence* (SI) [7-8], i.e., the self-organized stigmergic-based [9-11] property of a complex system whereby the collective behaviors of (unsophisticated) entities interacting locally with their environment cause coherent functional global patterns to emerge [12]. This line of

research recognized as *Ant Colony Optimization* (ACO) [13-15], uses a set of stochastic cooperating ant-like agents to find good solutions, using self-organized *Stigmergy* [16-19] as an indirect form of communication mediated by an artificial pheromone, whereas agents deposit pheromone-signs on the edges of the problem-related complex network, encompassing a family of successful algorithmic variations such as: *Ant Systems* (AS) [20], *Ant Colony Systems* (ACS) [21], *Max-Min Ant Systems* (Max-Min AS) [22] and *Ant-Q* [23].

Albeit being extremely successful these algorithms mostly rely on positive feedbacks [13], causing excessive algorithmic exploitation over the entire combinatorial search space. This is particularly evident over well-known benchmarks as the symmetrical *Traveling Salesman Problem* (TSP) [24]. Being these systems comprised of a large number of frequently similar components or events, the main challenge is to understand how the components interact to produce a complex pattern that is still a feasible solution [25] (in our case study, an optimal robust solution for hard NP-complete dynamic TSP-like combinatorial problems).

In order to overcome this hard search space exploitation-exploration compromise, our present algorithmic approach follows the route of very recent biological findings [26-28] showing that forager ants lay attractive trail pheromones to guide nest mates to food, but where, the effectiveness of foraging networks were improved if pheromones could also be used to repel foragers from unrewarding routes. Increasing empirical evidences for such a negative trail pheromone exists, deployed by *Pharaoh's* ants (*Monomorium pharaonis*) as a 'no entry' signal to mark unrewarding foraging paths.

The new algorithm was exhaustively tested on a series of well-known benchmarks over hard NP-complete COP's, running on symmetrical TSP [24]. Different network topologies and stress tests were conducted over low-size TSP's, medium-size TSP's, and as well as large sized ones. We show that the new co-evolved stigmergic algorithm compared favorably against the benchmark. In order to deeply understand how a second co-evolved pheromone was useful to drive the collective system into such results, the influence of negative pheromone was tracked (fig. 3-4-5), and as in previous tests [29-30], a refined phase-space map was produced mapping the pheromones ratio between a pure Ant Colony System and the present second-order approach.

2 Towards a Co-evolving Swarm-Intelligence

In order to make use of co-evolution we created a double-pheromone model on top of the traditional ACS, thus allowing the comparison between the two, by having an additional parameter. Traditional approaches to the TSP via Ant Systems include only a positive reinforcement pheromone. Our approach uses a second negative pheromone, which acts as a marker for forbidden paths. These paths are obtained from the worse tour of the ants and this pheromone then blocks access of ants in subsequent

tours. This blockade isn't permanent and as the pheromone evaporates it allows paths to be searched again for better solutions. This leads to equations 5-9 that expand equations 1-4 of the original ACS and AS approaches.

Ant Colony System (ACS, [21]) state transition rule

$$s = \begin{cases} \operatorname{argmax}_{u \in J_k(r)} \{ [\tau(r, u)] \cdot [\eta(r, u)]^\beta \}, & \text{if } q \leq q_0 \text{ (exploitation)} \\ S, & \text{otherwise (biased exploration)} \end{cases} \quad (1)$$

Ant System (AS, [20]) random proportional rule

$$p_k = \begin{cases} \frac{[\tau(r, s)] \cdot [\eta(r, s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r, u)] \cdot [\eta(r, u)]^\beta}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Ant Colony System (ACS, [21]) local updating rule

$$\tau(r, s) \leftarrow (1 - \rho) \cdot \tau(r, s) + \rho \cdot \Delta \tau(r, s) \quad (3)$$

Ant Colony System (ACS, [21]) global updating rule

$$\tau(r, s) \leftarrow (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta \tau(r, s) \quad (4)$$

2.1 ACS Double-Pheromone State Transition Rule

Following the guidelines of Dorigo and Gambardella [21], in ACS the state transition rule is as follows: an ant positioned on node r chooses the city s to move to by applying the rule given in Eq.5

$$s = \begin{cases} \operatorname{argmax}_{u \in J_k(r)} \{ [\tau^+(r, s)]^\alpha \cdot [\eta(r, s)]^\beta \cdot [\tau^-(r, s)]^{\alpha-1} \}, & \text{if } q \leq q_0 \\ S, & \text{otherwise} \end{cases} \quad (5)$$

where q is a random number uniformly distributed in $[0...1]$, q_0 is a parameter ($0 < q_0 < 1$) and S is a random variable selected according to the probability distribution in Eq. 6. In Eq.5, the parameter q_0 determines the relative importance of exploitation versus exploration, that is, every time an ant in city r has to choose a city s to move to, it samples a random number between $0 < q < 1$. If $q < q_0$ then the best edge according to Eq.5 is chosen (exploitation), otherwise an edge is chosen according to Eq.6 (biased exploration) or *random-proportional rule* coming from the classic *Ant System (AS)*, which follows:

$$p_k = \begin{cases} \frac{[\tau^+(r,s)]^\alpha \cdot [\eta(r,s)]^\beta \cdot [\tau^-(r,s)]^{\alpha-1}}{\sum_{u \in J_k(r)} [\tau^+(r,u)]^\alpha \cdot [\eta(r,u)]^\beta \cdot [\tau^-(r,u)]^{\alpha-1}}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

Eq.6 gives the probability with which ant k in city r chooses to move to city s , where τ is the pheromone on the (r,s) network edge, $\eta=1/\delta$ is the inverse of the distance $\delta(r,s)$, $J_k(r)$ is the set of cities that remain to be visited by ant k positioned on city r (in order to make the solution feasible), and β is a parameter which determines the relative importance of pheromone versus distance ($\beta>0$) and α controls the ratio between positive and negative pheromone influences. In Eq.6 the pheromones on edge (r,s) are multiplied by the corresponding heuristic value $\eta(r,s)$, thus favoring the choice of edges which not only are shorter but also with a greater amount of positive pheromone and some amount of negative pheromone.

The final ACS state transition rule resulting from Eqs. 5 and 6 is then called *pseudo-random-proportional rule*. This state transition rule, as with the previous AS random-proportional rule, favors transitions towards nodes connected by short edges and with a large amount of pheromone.

2.2 ACS Double-Pheromone Global Updating Rule

While AS used L_k , the length of the tour performed by every ant k , as a heuristic measure for the pheromone global updating rule, ACS instead focus only in the globally best ant, among all m , i.e. the ant which constructed the shortest tour from the beginning of the trial is allowed to deposit pheromone. This choice, along with the use of the *pseudo-random-proportional* state transition rule (above) was intended to make the search more directed: ants search in a neighborhood of the best tour found up to the current iteration of the algorithm. Global updating is performed after all ants have completed their tours. The pheromone level is then updated by applying the global updating rule of Eq.7 and 8 below:

$$\begin{aligned} \tau^+(r,s) &\leftarrow (1-\rho^+) \cdot \tau^+(r,s) + \rho^+ \cdot \Delta\tau^+(r,s) \\ \Delta\tau^+(r,s) &= \begin{cases} (L_{gb})^{-1}, & \text{if } (r,s) \in \text{Global best tour} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (7)$$

where $0 < \rho^\pm < 1$ is the pheromone decay parameter (evaporation) and L_{gb} the length of the globally best tour from the beginning of the trial. As it was the case in AS, the ACS global pheromone updating provides a greater amount of pheromone to shorter tours. Eq.7 dictates that only those edges belonging to the global best tour will receive reinforcement while Eq.8 dictates that only those edges that belong to the worse tour receive negative pheromone deposition.

2.3 ACS Double-Pheromone Local Updating Rule

In order for the 2nd order algorithm (as in ACS) to build a solution, i.e. a TSP tour, ants visit edges and change their pheromone level by applying a local updating rule given by Eq.9:

$$\begin{aligned} \tau^-(r,s) &\leftarrow (1-\rho^-) \cdot \tau^-(r,s) + \rho^- \cdot \Delta\tau^-(r,s) \\ \Delta\tau^-(r,s) &= \begin{cases} (nL_{gb})^{-1}, & \text{if } (r,s) \in \text{Global worse tour} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

$$\begin{aligned} \tau^+(r,s) &\leftarrow (1-\rho^+) \cdot \tau^+(r,s) + \rho^+ \cdot \Delta\tau^+(r,s) \\ \tau^-(r,s) &\leftarrow (1-\rho^-) \cdot \tau^-(r,s) + \rho^- \cdot \Delta\tau^-(r,s) \end{aligned} \tag{9}$$

where $0 < \rho < 1$ is a parameter. From here several options are possible where, $\Delta\tau(r,s)$ could assume the form of $\Delta\tau(r,s) = \gamma \cdot \max \tau(s,z) [z \in J_k(S)]$ similarly to a reinforcement learning problem, onto which ants have to learn which city to move to as a function of their current location. This first option assumes Q -learning, an algorithm which allows an agent to learn such an optimal policy by the recursive application of a rule similar to that in Eq.4, giving rise to the first Ant- Q ant systems. In fact, $\Delta\tau(r,s) = \gamma \cdot \max \tau(s,z)$ is exactly the same formula used in Q -learning where $0 < \gamma < 1$ is a parameter. The other two choices are normally: (1) setting $\Delta\tau(r,s) = \tau_0$, being τ_0 the initial pheromone level, or (2) simply setting $\Delta\tau(r,s) = 0$. Finally, experiments could also be ran in which local-updating are not applied at all, that is, where the local updating rule is not used as in the case of the older and previous AS).

Current research work however, suggests that local-updating is not only definitely useful, but that the pheromone local updating rule with $\Delta\tau(r,s) = 0$ yields worse performance than $\Delta\tau(r,s) = \tau_0$ or even Ant- Q . In fact, $\Delta\tau(r,s) = \tau_0$ was chosen for the standard ACS, from the beginning [13-15][20,21,23]. Since the ACS local updating rules not only requires less computation than Ant- Q as well as achieving better results, we chose to focus our attention on ACS, which will be used, along others, to run the comparison experiments against our new co-evolved pheromone-based algorithm in the following paper sections.

3 Results

The new algorithm was exhaustively tested on a series of well-known benchmarks over hard NP-complete COP's) running on symmetrical TSP's. Different network topologies and stress tests were conducted over low-size TSP's (eil51.tsp; eil78.tsp; kroA100.tsp), medium-size (d198.tsp; lin318.tsp; pcb442.tsp; rat783.tsp) as well as large sized ones (fl1577.tsp; d2103.tsp) [numbers here referring to the number of nodes in each network].

Table 1. Comparison of Standard ACS with the 2nd order AS algorithm

<i>problem</i>	<i>n.º of nodes</i>	<i>Standard ACS</i>	<i>2nd order⁺ AS</i>	<i>optimal tour</i>
eil51.tsp	51	427.96	428.87	426
eil78.tsp	78	**	544.65	538
kroA100.tsp	100	21285.44	21285.44	21282
d198.tsp	198	16054	15900.2	15780
lin318.tsp	318	42029***	42683.90	42029
pcb442.tsp	442	51690	51464.48	50778
rat783.tsp	783	9066	8910.48	8806
fl1577.tsp	1577	23163	22518	22249
d2103.tsp	2103	-	81151.9	80450

All optimal tours from <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/STSP.html>

+ Average over 20 runs and limited to 1000 iterations

** Value for similar problem eil75,.tsp - 542.37 *** uses 3-opt local search

Comparing Traditional AS Models with 2nd Order

It is clear from table 1 that the 2nd order AS performs at least equally, if not better, than the standard ACS. It is clearly seen the averages of the runs (bold) that are better than the traditional ACS.

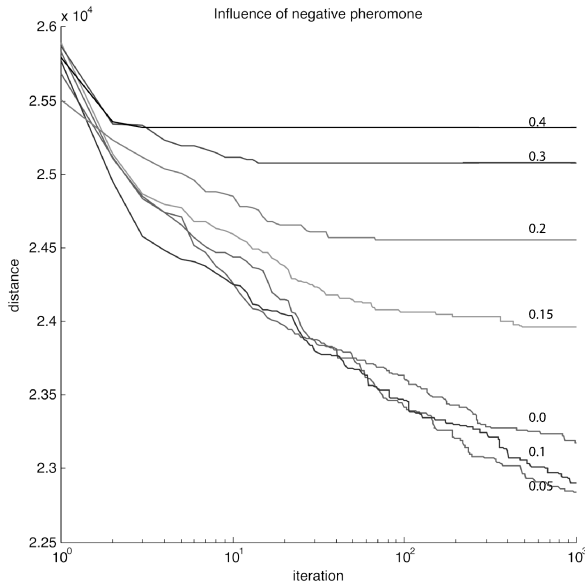


Fig. 1. Influence of negative pheromone on *kroA100.tsp* problem (values on lines represent 1-ALPHA)

We investigated the evolution of different ratios of negative pheromone and found that a small amount of negative pheromone applied as a non-entry signal indeed produces better results, but the effect is cancelled if the ratio of the negative pheromone is high when compared to the positive pheromone.

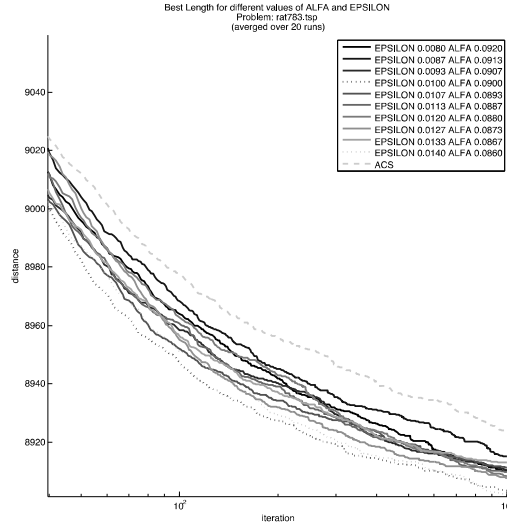


Fig. 2. Best tour of the 2nd order AS for different ratios of negative pheromone in the *rat783.tsp* problem

The effect of negative pheromone can be observed both in figure 1 and figure 2 where one can observe that small amounts of negative pheromone produce better results and quicker convergence to those results. On the other hand if one increases the ratio of negative pheromone to higher values then it isn't possible to ripe the benefits of the no-entry signal and the system performs worse.

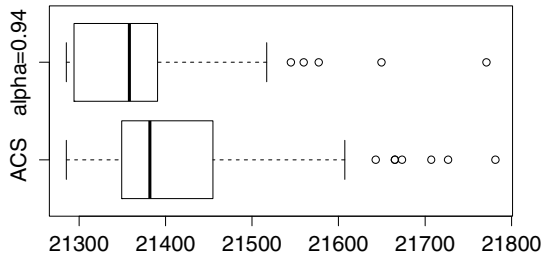


Fig. 3. Influence of negative pheromone on *kroA100.tsp* problem

The detailed analysis of the *kroA100.tsp* problem showed that this effect is statistically significant. Comparing 120 runs with $\alpha=1$ (equivalent to traditional ACS) and $\alpha=0.94$, we obtained a p -value of 3×10^{-4} . This result is summarized in figure 3, where one compares traditional ACS with our 2nd order approach.

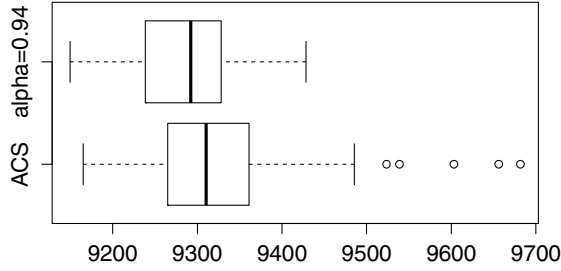


Fig. 4. Influence of negative pheromone on *rat783.tsp* problem

The same results were observed for problem *rat783.tsp* when comparing 70 runs of the ACS ($\alpha=1$) with 70 runs of the 2nd order approach (with $\alpha=0.94$) in figure 4. The two samples means were tested for statistical significance resulting in a p -value of 2.2×10^{-3} .

Both these examples show that on average the 2nd order approach performs better than traditional ACS. This effect of the negative pheromone is important but cannot be extended further as to dominate the solving strategy, making results worse. This can be seen clearly on figure 5 where further diminishing of α (giving more weight to negative pheromone as a consequence) produces worse results.

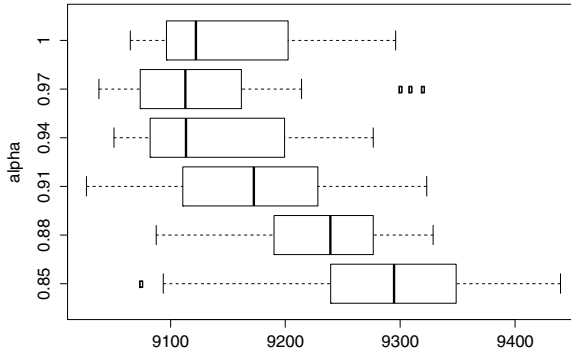


Fig. 5. If dominant, negative pheromone has negative impact (problem *rat783.tsp*)

4 Conclusion

We show that the new co-evolved stigmergic algorithm compared favorably against the benchmark. The inclusion of a negative pheromone acting as a ‘non-entry’ signal in the strategy of construction of solutions is beneficial as the convergence to optimal solutions is quicker, as shown in figure 2, while achieving better results (figures 3 and 4). The algorithm was able to equal or majorly improve every instance of those standard algorithms.

The new algorithm comprises a second order approach to *Swarm Intelligence*, as pheromone-based no entry-signals cues, were introduced, coevolving with the standard pheromone distributions (collective cognitive maps [12]) in the aforementioned known algorithms.

The use of the negative pheromone is limited to small quantities (*alpha* close to 1, but not 1, in which case we would end up with a pure ACS) and cannot be extended to a point of dominance in the search strategy as shown in figure 5. The results found for the TSP problems in that case are severely worse. This implies that the use of a negative pheromone strategy has to be fine tuned as not to dominate the search strategy. This is done with the introduction of the parameter *alpha* that balances the weight of the two pheromones deposition in equations 5 and 6.

This work has implications in the way large combinatorial problems are addressed as the double feedback mechanism shows improvements over the single-positive feedback mechanisms in terms of convergence speed and of major results.

References

1. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, MI (1975)
2. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, USA (1989)
3. Fogel, D.B.: *Evolutionary Computation*. IEEE Press, Piscataway (1995)
4. Siarry, P., Michalewicz, Z.: *Advances in Metaheuristics for Hard Optimization*. Springer (2008)
5. Gonzalez, T.F. (ed.): *Approximation Algorithms and Metaheuristics*. CRC Press (2007)
6. Alba, E.: *Parallel Metaheuristics. A New Class of Algorithms*. Wiley, Cambridge (2005)
7. Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence: From Natural to Artificial Systems*. Santa Fe Institute series in the Sciences of Complexity. Oxford Univ. Press, New York (1999)
8. Blum, C., Merkle, D. (eds.): *Swarm Intelligence: Introduction and Applications*. Natural Computing Series. Springer, Heidelberg (2008)
9. Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G., Bonabeau, E.: *Self-Organization in Biological Systems*. Princeton University Press, Princeton (2003)
10. Chialvo, D.R., Millonas, M.M.: How Swarms build Cognitive Maps. In: Steels, L. (ed.) *The Biology and Technology of Intelligent Autonomous Agents*. NATO ASI Series, vol. 144, pp. 439–450 (1995)
11. Millonas, M.M.: A Connectionist-type model of Self-Organized Foraging and Emergent Behavior in Ant Swarms. *J. Theor. Biol.* 159, 529 (1992)
12. Ramos, V., Fernandes, C., Rosa, A.C.: On Self-Regulated Swarms, Societal Memory, Speed and Dynamics. In: Rocha, L.M., Yaeger, L.S., Bedau, M.A., Floreano, D., Goldstone, R.L., Vespignani, A. (eds.) *Artificial Life X - Proc. of the Tenth Int. Conf. on the Simulation and Synthesis of Living Systems*, Bloomington, Indiana, USA, pp. 393–399. MIT Press (2006)
13. Dorigo, M., Maniezzo, V., Colomi, A.: Positive Feedback as a Search Strategy, Technical report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy (1991)
14. Dorigo, M., Di Caro, G.: The Ant Colony Optimization Metaheuristic. In: Corne, D., Dorigo, M., Glover, F. (eds.) *New Ideas in Optimization*, p. 11. McGraw-Hill, New York (1999)

15. Dorigo, M., Di Caro, G., Gambardella, L.M.: Ant algorithms for Discrete Optimization. *Artificial Life* 5(2), 137 (1999)
16. Grassé, P.P.: La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. La théorie de la Stigmergie: Essai d'interprétation des termites constructeurs. *Insect Sociaux* 6, 41–83 (1959)
17. Theraulaz, G., Bonabeau, E.: A Brief History of Stigmergy. *Artificial Life, Special Issue Dedicated to Stigmergy* 5(2), 97–116 (1999)
18. Abraham, A., Grosan, C., Ramos, V.: *Stigmergic Optimization*. SCI, vol. 31. Springer, Heidelberg (2006)
19. Diaf, M., Hammouche, K., Siarry, P.: From the Real Ant to the Artificial Ant. In: *Nature-Inspired Informatics for Intelligent Applications and Knowledge Discovery*, pp. 298–322 (2010)
20. Dorigo, M., Maniezzo, V., Colomni, A.: Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Trans. Syst., Man, and Cybern. - Part B* 26(1), 29 (1996)
21. Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning approach to the Travelling Salesman Problem. *IEEE Trans. Evol. Computation* 1(1), 53 (1997)
22. Stützle, T., Hoos, H.H.: MAX-MIN Ant System. *Future Generation Comput. Syst.* 16(8), 889 (2000)
23. Gambardella, L.M., Dorigo, M.: Ant-Q: A Reinforcement Learning Approach to the Travelling Salesman Problem. In: Prieditis, A., Russell, S. (eds.) *Proceedings of the Twelfth International Conference on Machine Learning, ML 1995, Tahoe City, CA*, pp. 252–260. Morgan Kaufmann (1995)
24. Lawler, E.L., Lenstra, J.K., Rinnooy-Kan, A.H.G., Shmoys, D.B.: *The Travelling Salesman Problem*. Wiley, New York (1985)
25. Ramos, V., Almeida, F.: Artificial Ant Colonies in Digital Image Habitats: A Mass Behavior Effect Study on Pattern Recognition. In: Dorigo, M., Middendorf, M., Stützle, T. (eds.) *From Ant Colonies to Artificial Ants – ANTS 2000 - 2nd Int. Wkshp on Ant Algorithms*, pp. 113–116 (2000)
26. Robinson, E.J.H., et al.: Insect communication - 'No entry' signal in ant foraging. *Nature* 438(7067), 442 (2005)
27. Robinson, E.J.H., Jackson, D., Hocombe, M., Ratnieks, F.L.W.: No entry signal in ant foraging (Hymenoptera: Formicidae): new insights from an agent-based model. *Myrmecological News* 10, 120 (2007)
28. Grüter, C., Schürch, R., Czaczkes, T.J., Taylor, K., Durance, T., et al.: Negative Feedback Enables Fast and Flexible Collective Decision-Making in Ants. *PLoS ONE* 7(9), e44501 (2012), doi:10.1371/journal.pone.0044501
29. Rodrigues, D.M.S., Louçã, J., Ramos, V.: From Standard to Second-Order Swarm Intelligence Phase-space Maps. In: Thurner, S. (ed.) *8th European Conference on Complex Systems*, poster, Vienna, Austria (September 2011)
30. Ramos, V., Rodrigues, D.M.S., Louçã, J.: Spatio-Temporal Dynamics on Co-Evolved Stigmergy. In: Thurner, S. (ed.) *8th European Conference on Complex Systems*, poster, Vienna, Austria (September 2011)

Hybrid Approach Using Rough Sets and Fuzzy Logic to Pattern Recognition Task

Andrzej Zolnierek and Marcin Majak

Department of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

Abstract. In this paper a hybrid classifier construction using rough sets and fuzzy logic is presented. Nowadays, we tackle with many realistic multi-dimensional problems with continuous values and overlaps in the feature space which require sophisticated recognition algorithms. Many methods have been proposed in the literature to improve classification accuracy, but it is increasingly harder to build new classifier from the scratch. Instead, new fusion methods are proposed to overcome this problem. In our rough-fuzzy approach data pre-processing and crisp discretization have a significant impact on the final classification efficiency. To deal with the problem of finding the optimal cuts in the feature space a genetic algorithm was proposed. After the algorithm description, in this paper also simulation investigations using different datasets from UCI Machine Learning Repository are presented.

Keywords: Rough Sets, Fuzzy Logic, Hybrid Pattern Recognition Algorithm.

1 Introduction

In the recent years algorithms of data mining have attracted great attention. Rule-based systems have been widely used in control problems and recently have been successfully applied in classification problems. One can find many algorithm propositions in the literature, but lately the abilities of single classifier has been significantly exhausted. Now, hybrid algorithm constructions gain greater appeal [1], [2]. Classifiers with different capabilities are merged to improve the classification accuracy. In this article fuzzy logic and rough sets are combined to build hybrid classifier used in pattern recognition task.

Interpretability and comprehensibility are the main advantages of fuzzy and rough sets algorithms. On the other hand, creation of easy to understand IF-THEN rules becomes challenging in the high dimensional problems. This is a transaction between the number of rules and decision how accurately we want to describe the concept. In the past few years the most common and emerging areas of computational intelligence were rough sets, fuzzy sets, genetic algorithms and different kinds of hybrid combiners [3], [4], [5], [6] which give the most promising results.

The rough sets theory, introduced by Pawlak in 1982 was a new method of the uncertainty modeling in the complex pattern recognition problems without

any statistical knowledge about the probability distributions region [7]. Another possibility of using this theory was attribute selection. In rough sets theory reasoning is based on rules which are induced from the available knowledge contained in the training set. These rules can be either certain or possible which is the consequence of using lower and upper set approximations describing the target set. In case of rough sets, choosing the proper discretization step of the input data is a key point in algorithm design. If we choose very small step of granulation then we can separate classes very precisely, but on the other hand due to the shortage of training data it can happen that some generated rules are dummy, because we cannot find any representative patterns fulfilling the rule. Going to the opposite side and choosing greater step of granulation causes that more patterns are indiscernible from the point of available information [8].

Next to rough sets, fuzzy logic based systems are also commonly applied in many recognition problems. This method deals with reasoning which is approximate rather than fixed and exact. An element membership to the set is described by the membership function μ instead of well-known Boolean logic where an element can only fully belong to the set (true) or is totally exclusive (false). Fuzzy algorithm (the same as rough sets) is able to process incomplete data and provide approximate solutions. Creation of proper IF-THEN rules is the main challenge in fuzzy logic. One has to ensure that rules should be simple, comprehensive and covering possibly the greatest part of feature space. The basic approach assumes that each attribute is divided into the same number of intervals and each interval is assigned with one membership function μ . Single rule is a permutation of μ from each attribute. This solution is applicable only for trivial problems, but for high dimensional ones it is not possible to generate proper number of rules to fully cover attribute space. Some methods of attribute selection and heuristic algorithms must be applied [1], [2], [9]. In this work a genetic algorithm is used to obtain effective subset of rules.

To sum up, the aim of this paper is to present new hybrid classifier which is built using rough sets, fuzzy logic and genetic algorithm. This is a two stage classifier where on the first level recognition decision is made by rough sets classifier. When no certain or possible rule is activated at this stage, then incoming pattern is directed to the next stage - fuzzy logic classifier. The main motivation of this work was to create new classifier which would be able to perform well with different dataset. Moreover, in the previous work we experienced that rough sets and fuzzy logic cannot handle satisfactorily with highly dimensional datasets and time for generation of IF-THEN rules was not acceptable. Due to these reasons we decided to introduce in this work genetic algorithm to reduce feature space i.e. the set of rules on both stages of classification.

The organization of the paper is as follows: in the Section 2 we present the problem statement connected with pattern recognition area. Then, Section 3 describes in greater details algorithm construction based on rough sets theory and fuzzy logic. In the Section 4 the results of tests using some dataset from UCI Machine Learning Repository are presented. Section 5 concludes the paper with final results remarks and presents further application improvements.

2 Problem Statement

Let us assume that the pattern is in the state $j \in M$, where M is an m -element set of possible states numbered with the successive natural numbers. The state j is unknown and does not undergo the direct observation. What we can only observe are features or attributes by which a state manifests itself. We will denote a d -dimensional measured feature vector by $x \in X$. Generally, in order to classify unknown object we assume that we have training set consisting of N training patterns:

$$S = (x_1, j_1), (x_2, j_2), \dots, (x_N, j_N) \tag{1}$$

where x_α, j_α denote d -dimensional pattern and its assigned true classification, respectively. In general the decision algorithm with learning procedure should use every time as well observed data i.e. the feature vector as the knowledge included in the training set. In consequence, the algorithm with learning set is of the following form:

$$j = \Psi(S, x), \quad j \in M \tag{2}$$

The training set can be viewed as table where each column represents measured attribute for the pattern and each row stands for the particular object. The knowledge stored in the information table has a granular structure which causes that some objects become undiscerned. In the information system we can define in a different way the subset $C \subset A$ of condition attributes and the single-element set $D \subset A$ which will be the decision attribute. Taking into account the set of condition attributes C , let us denote by P_j the subset of patterns from U (the universe of discourse which is a finite set of objects) for which the decision attribute is equal to $j, j = 1, \dots, m$. Then, for every j we can defined the C -lower approximation $C_*(P_j)$ and the C -upper approximation $C^*(P_j)$ of set P_j , respectively [7], [8]. Consequently, we can define C -boundary region of P_j as follows:

$$CN_B(P_j) = C^*(P_j) - C_*(P_j) \tag{3}$$

If for any j the boundary region $CN_B(P_j)$ is the empty set then P_j is *crisp*, while in the opposite case we deal with *rough set*. For every decision system we can formulate its equivalent description in the form of set of decision rules $For(C)$. Each row of the decision table will be represented by a single IF-THEN formula, where on the left side of this implication we have logical product (and) of all expressions from C such that every attribute is equal to its value. On its right side we have decision attribute which is equal to the particular value from M . These formulas are necessary for constructing final rough sets pattern recognition algorithm.

Fuzzy rule-based classification systems represent another type of reasoning based on the degree of membership. It is a superset of Boolean logic that has been extended to handle the concept of partial truth- values between "completely true" and "completely false". In the literature one can find many methods for fuzzy algorithm construction [5], [10]. Generally, for the d -dimensional classification problem fuzzy classifier is based on IF-THEN rules where single rule

consists of many appropriate antecedents and one consequent determining class affiliation. Each antecedent and consequent is described by linguistic variable-words or sentences from a natural language. An exemplary rule is given by (4).

$$R_q : IF(x^1 = A_{q_1}^1) \text{ AND } (x^2 = A_{q_2}^2) \dots \text{ AND } (x^d = A_{q_d}^d) \text{ THEN class=j} \quad (4)$$

where x^i is the value of i -th attribute, $A_{q_i}^i$ is the linguistic variable for i -th attribute. Linguistic variables describe rules and facts, i.e a linguistic variable such as height may have the following representation: small, medium, high. They were introduced into fuzzy logic to make rules more readable. On the other hand, a membership function is used to quantify a linguistic term, describing how variable can belong to multiple sets at the same time.

3 Pattern Recognition Algorithm

In this section we will present two stage hybrid classifier in greater details. This paper is a continuation of authors works connected with rough sets [11], [12], [13] but here we extend it and additionally apply fuzzy logic and genetic algorithm. In mentioned papers designed algorithm assumed the same number of intervals for every feature and what is more important all attributes were used in classification process which made the recognition very complicated for more complex problems. In this paper each attribute is discretized independently and to find those partitions we propose a genetic algorithm. Additionally, in cases when rough sets algorithm cannot classify a pattern then fuzzy logic classifier is applied. The description of our classifier is presented below:

1. Stage 1 - application of rough sets algorithm.

This algorithm is constructed according to the following steps:

- (a) In the first step we look for the optimal cuts of each attribute in rough sets algorithm with the help of genetic algorithm. The previous implementation used an arbitrary chosen step of granulation and its classification efficiency strongly depends on it. What is more, each attribute was discretized with the same pre-defined interval. Very often we do not need to use all attributes for correct classification. Some of them are very meaningful, so should be approximated more precisely, while others contain noise and only deteriorate classification. Generally, finding the optimal cuts in the feature space and applying attribute selection is a NP-hard task, so to overcome this problem we apply genetic algorithm. The length of each individual in the population is the same as the number of attributes characterizing pattern for a particular dataset. Each allele in the chromosome encodes information about potential cuts of corresponding attribute. To enable feature selection an extra variable *don't use* is introduced which indicates that a given attribute is excluded from the formulation of IF-THEN rules. For clarity, let consider an individual describing discretization of four dimensional feature space: $|don't\ use|5|3|don't\ use$. This encoding means that the first and

the fourth attributes are not taken into account in the classification (*don't use* variable is used), the second attribute is divided into five even intervals and the third into three regions. The fitness function used for assessing each individual is given by (5)

$$F = w_1 \cdot NR - w_2 \cdot NNR + NR \cdot \left(\frac{1}{NOA}\right)^2 \tag{5}$$

where w_1, w_2 are empirically chosen weights (fulfilling the obvious condition $w_1 < w_2$), NR is the number of correctly classified patterns using partition encoded in the individual, NNR is the number of misclassified objects and NOA represents the number of attributes which are taken into account in the current classification phase. For the example above, NOA would be 2. This fitness function takes into account not only the number of correctly recognized object, but prizes those individuals which are able to reject some attributes obtaining high classification accuracy at the same time. In each generation every individual from population is used to construct granulation for feature space and based on the classification from testing data the value of F is calculated. An individual with the maximum fitness indicator is used as the final solution. Initial population is generated randomly with an arbitrary chosen vector of numbers $K_l, (l = 1, \dots, d)$, corresponding to the initial step of granulation. Final parameters for genetic algorithm and fitness function were chosen empirically after many tests which were performed for different population sizes and weights in function F . After these tests we have chosen $w_1 = 2, w_2 = 5$ while the final genetic settings were: population size 20, generations 100, mutation probability 0.1, cross-over probability 0.9. Of course initial settings affects our hybrid algorithm accuracy, so we conducted many tests to check which paramters are the best to obtain satisfactory results in reasonable algorithm execution time. Table 1 shows how initial settings affects hybrid classifier accuracy for an exemplary Pima dataset (C-accuracy in percentage, S-standard deviation in percentage).

Table 1. Impact of genetic algorithm's parameters on hybrid classifier accuracy

Population Size	Generations	C	S
5	50	71.6	7.2
20	50	77.6	8.0
100	50	76.9	9.0
5	100	73.4	6.5
20	100	81.3	3.2
100	100	80.9	3.1
5	100	74.9	3.2
20	100	80.6	3.6
100	100	80.6	3.1

- (b) Next step is based on granulation pre-processing for those attributes chosen by the best chromosome obtained in step 1. After the granulation procedure value of each pattern's attribute is discretized and represented by the appropriate number of interval in which this attribute is included. For further usage let denote the l -th attribute and its p_l -th value or interval ($p_l = 1, \dots, K_l$) by $v_{p_l}^l$.
- (c) In this step, using available training data we generate the set $For(C)$ of all decision formulas according to the granulation from the previous stage:

$$\text{IF } (x^1 = v_{p_1}^1) \text{ AND } \dots \text{ AND } (x^d = v_{p_d}^d) \text{ THEN } \Psi(S, x) = j \quad (6)$$

During the learning procedure each rule is given the strength factor which is the fraction of correctly classified patterns over all objects which activated this rule.

- (d) In the next step for the set of formulas $For(C)$, for every $j = 1, \dots, m$ we calculate C -lower approximation $C_*(P_j)$, C -upper approximation $C^*(P_j)$ and boundary region $CN_B(P_j)$.
- (e) In the last step we can classify an incoming pattern by looking for matching rules in the set $For(C)$. At this step three possible actions can be undertaken. If there is only one matching rule, then we classify this pattern to the class which is indicated by its decision attribute j , because for sure such a rule belongs to the lower approximation of all rules indicating j . If there is more than one matching rule in the set $For(C)$, it means that the recognized pattern should be classified by the rules from the boundary region and in this case the final decision is determined by index of boundary region for which the strength of corresponding rules is maximal. When none rule is activated or there are two or more classes with the same strength factor, then pattern is rejected and will be classified in the next stage by fuzzy logic classifier.

2. Stage 2 - application of fuzzy logic algorithm.

In the fuzzy logic algorithm construction the key point lies in defining correct membership functions set for each attribute. Where there is no expert knowledge at hand some data-mining methods must be used to extract fuzzy rules, i.e. histogram examination, heuristic algorithms [6]. Additionally one has to decide which type of membership function should be used. In the literature one can find many examples of artificial intelligence algorithms to generate fuzzy rules such as: genetic algorithm or neural networks. In the previous phase genetic algorithm was used to find optimal cuts in feature space for rough sets algorithm. It gave good results so we decided to apply it again, but this time in different context i.e. to produce fuzzy rules. As the input, triangular shape membership functions were used and their locations in the feature space were determined according to genetic algorithm working on the available training dataset. In the literature this procedure is called genetic-based machine learning algorithm. Before we describe fuzzy algorithm few assumptions have to be made: for the learning procedure N training patterns are available and a set L of linguistic values and their membership

functions are given for describing each attribute. The second condition is the most important and the proper location of membership functions affects the classification accuracy. It is required to explain in-depth how set L is generated, how to divide each attribute into linguistic values and finally how to describe each membership function. Below the steps for constructing fuzzy logic algorithm are presented:

- (a) In the first step each attribute is divided into the same number of intervals and for each interval new membership function is assigned. Division can be done in different ways, but in this paper the set of L was generated using typical triangular functions. In simulations carried out for this work the maximum number of membership functions per attribute was 14 (greater number of L does not ensure better results in training phase, but significantly extends algorithm execution). For each membership function from L unique linguistic variable is attached plus one additional variable *don't use* to indicate that the given attribute is not used in rule generation.
- (b) In the next step to produce fuzzy rules we applied genetic algorithm, which starts with $N = 20$ individuals. This time single individual encodes information about one rule which is in the form represented by equation (4). The length of the chromosome is the same as the number of attributes describing the pattern x and each allele has value determining which linguistic variable is used in the current rule. Through the mutation and cross-over operators new individuals are added to existing population and at the end of each generation $N = 20$ best individuals are chosen according to the fitness function F from eq. (5) to constitute next generation.
- (c) In the consequently step we assign class label to a single rule and compute its strength. For each training pattern x_p let calculate the compatibility grade of a single rule connected with its antecedent part $A_r = (A_{r1}, A_{r2}, \dots, A_{rd})$ using the product operator of each membership function $\mu_{A_{r_i}}$ determined for A_{r_i} :

$$\mu_{A_r}(x_p) = \mu_{A_{r1}}(x_p) \cdot \mu_{A_{r2}}(x_p) \cdot \dots \cdot \mu_{A_{rd}}(x_p) \tag{7}$$

If we know how to calculate the compatibility grade of each training pattern then we can determine C_r and CF_r for each rule. The fuzzy probability $Pr(class\ j|A_r)$ of class j , $j = (1, \dots, m)$ indicating how pattern x can be associated with class j is shown below [2]:

$$Pr(class\ j|A_r) = \frac{\sum_{x_p \in class\ j} \mu_{A_r}(x_p)}{\sum_{p=1}^m \mu_{A_r}(x_p)} \tag{8}$$

For the r -th rule R_r the label of class is assigned according to the winning rule, which means that the label with maximal probability is chosen:

$$R_r : C_r = max_{j \in \{1, \dots, m\}} \{Pr(class\ j|A_r)\} \tag{9}$$

In the learning phase it can happen that rule R_r can be activated by patterns coming from different classes. To ensure the proper classification, each rule has a strength factor which tells how precisely rule R_r predicts the consequent class j .

$$R_r : CF_r = Pr(class\ j|A_r) - \sum_{j=1, j \neq C_r}^m Pr(class\ j|A_r) \quad (10)$$

If CF_r in (10) is negative then rule R_r is denoted as *dummy* and is not taken for further reasoning, otherwise it is used in defuzzification process to determine the final class label. Let assume that N_{rule} fuzzy rules are generated with indicators C_r , CF_r determined by (9), (10).

(d) Then in the last step the process of classification can be done as follows:

$$\Psi(S, x_p) = C_q \leftarrow \max_{j \in \{1, \dots, m\}} \{\mu_{A_q}(x_p) \cdot CF_r\} \quad (11)$$

The label of the class for unknown pattern is determined by a winner rule R_w that has the maximum compatibility grade and the rule strength CF_r . If multiple fuzzy rules have the same maximum product μ_{A_r} but different consequent classes then the classification is rejected. The same action is taken if no fuzzy rule is compatible with the incoming pattern x_p . At the beginning we have tried other methods of aggregation for fuzzy sets, but presented approach gave the best results.

4 Results

To test the effectiveness of proposed algorithm well-known datasets from UCI Machine Learning Repository were chosen: iris (A, 4 attributes), Bupa-liver disorders (B, 6 attributes), Pima-diabetes (C, 8 attributes), Haberman-breast cancer (D, 3 attributes), wdbc (E, 32 attributes), Thyroid disease (F, 5 attributes), wine (G, 13 attributes). To generate training and testing sets 10-fold cross validation was used, where the data set is divided into $k = 10$ subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other $k - 1$ subsets are merged together to form a training set. Because of genetic algorithm random nature each test for k -th dataset was performed 20 times and table 2 represents averaged values. For the comparison purposes four different classifiers were trained and tested on the same datasets: C 4.5, 3-KNN (weighted distance measure), Maximum likelihood classifier (MCL), SVM classifier (rbf kernel). Notation used in results table is as follows: C - algorithm accuracy in percentage, S - standard deviation of algorithm accuracy in percentage, F - micro-averaged measure metrix which is equal to harmonic mean of recall (ρ) and precision (π) calculated for each class from M :

$$\pi = \frac{\sum_{i=1}^m TP_i}{\sum_{i=1}^m (TP_i + FP_i)}, \rho = \frac{\sum_{i=1}^m TP_i}{\sum_{i=1}^m (TP_i + FN_i)}, F = \frac{2 \cdot \pi \cdot \rho}{\pi + \rho}$$

where TP_i number of patterns classified correctly to class i , FP_i number of objects that do not belong to class i but are assigned to class i , FN_i number of patterns not assigned to class i but actually belonging to class i . NA - averaged number of attributes used by hybrid algorithm in classification procedure.

Table 2. Classification accuracy

	C4.5			3-NN			SVM			MCL			Hybrid			
	C	S	F	C	S	F	C	S	F	C	S	F	C	S	F	NA
A	92.0	6.5	0.92	96.7	3.3	0.97	66.7	0.0	0.67	98.0	3.1	0.98	98.7	2.7	0.99	2.4
B	68.6	7.3	0.69	65.8	7.1	0.66	68.5	6.5	0.68	57.9	6.5	0.58	77.1	5.4	0.78	2.6
C	74.9	3.8	0.58	68.5	3.5	0.68	70.6	1.9	0.71	73.7	3.5	0.74	81.6	3.2	0.82	3.0
D	72.0	4.5	0.72	71.9	7.2	0.72	73.2	2.6	0.73	75.8	1.9	0.76	79.4	2.2	0.79	2.0
E	93.3	2.2	0.93	93.1	2.2	0.93	93.1	3.0	0.93	95.4	1.8	0.95	94.9	2.0	0.95	5.1
F	87.4	5.3	0.87	94.4	4.0	0.94	94.4	4.0	0.94	95.8	3.3	0.96	98.6	2.1	0.99	2.5
G	88.9	8.9	0.89	70.1	11.4	0.71	93.8	5.3	0.94	99.4	1.8	0.99	100.0	0.0	1.00	2.9

Results in table 2 show that for six out of seven datasets our hybrid classifier gives better results of accuracy comparing to other reference methods. By combining two classifiers we increased the possibility of correct classification because when rough sets algorithm rejects pattern there is a chance that fuzzy logic classifier will tackle with classification problem. Additionally, significant improvement was done by applying genetic algorithm in the pre-processing phase. First of all, problem dimensionality is reduced by introducing *don't use* variable which speeds up algorithm execution and IF-THEN rules are simpler and less complex. As an example consider wine dataset. In the standard rough sets algorithm we would use all 13 attributes with the same number of intervals K_l . Using our algorithm we managed to correctly classify 100% of testing objects only with three attributes on average. Irrelevant features were removed in every case which is shown by NA column in table 2.

5 Conclusions

In this paper new hybrid two stage classification algorithm is presented and compared with other classifiers. On the first stage the rough sets approach is applied for creation of decision rules while on the second stage fuzzy logic rules are induced. Proposed algorithm for both stages uses genetic algorithm in initial pre-processing phase for finding proper intervals in each attribute in case of rough sets approach and for features selection in case of fuzzy logic approach, respectively. New pattern can be classified by rough sets algorithm if in the set $For(C)$ certain or possible rule is activated, otherwise fuzzy logic is applied. In order to investigate the quality of this hybrid method several tests were performed using database from UCI Machine Learning Repository. Presented results confirms usefulness of this approach, but of course more profound tests and investigations are needed to compare with other results presented in the literature.

In the future work we would like to experiment with another types of hybridization connected with feature selection algorithms. In presented work we managed to accomplish two goals: feature reduction and high classification accuracy.

Acknowledgements. The work was supported by the statutory funds of the Department of Systems and Computer Networks, Wroclaw University of Technology, Poland.

References

- [1] Ishibuchi, H., Yamamoto, T.: Hybridization of fuzzy gbml approaches for pattern classification problem. *IEEE Trans. on Systems, Man, and Cybernetics* 35(2), 359–365 (2005)
- [2] Qinghua, H., Zongxia, X.: Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation. *Pattern Recognition* 40, 3509–3521 (2007)
- [3] Hu, Q., Shuang, A.: Robust fuzzy rough classifiers. *Fuzzy Sets and Systems* 183, 26–43 (2011)
- [4] Roy, A., Pal, S.: Fuzzy discretization of feature space for a rough set classifier. *Pattern Recognition Letters* 24(6), 895–902 (2003)
- [5] Shen, Q., Chouchoulas, A.: A rough-fuzzy approach for generating classification rules. *Pattern Recognition* 35, 2425–2438 (2003)
- [6] Wu, Q., Wang, T., Ji-Sheng, L.: New research on fuzzy rough sets. In: *Proceedings of the Fifth International Conference on Machine Learning and Cybernetics*, pp. 13–16 (2006)
- [7] Slowinski, R.: *Intelligent decision support: handbook of applications and advances of the rough sets theory*. Kluwer Academic Publishers, Dordrecht (2010)
- [8] Pawlak, Z.: Rough sets, decision algorithms and bayes theorem. *European Journal of Operational Research* 136(1), 181–189 (2002)
- [9] Khoo, L., Zhai, L.: A prototype genetic algorithm-enhanced rough set-based rule induction system. *Computers in Industry* 46, 95–106 (2001)
- [10] Mendes, R.R.F., Voznika, F.D.B., Freitas, A.A., Nievola, J.C.: Discovering fuzzy classification rules with genetic programming and co-evolution. In: Siebes, A., De Raedt, L. (eds.) *PKDD 2001. LNCS (LNAI)*, vol. 2168, pp. 314–325. Springer, Heidelberg (2001)
- [11] Kurzynski, M., Zolnierek, A.: Rough sets and fuzzy sets theory applied to the sequential classification - algorithms and applications. *Polish Journal of Environmental Studies* 17(2B), 68–77 (2008)
- [12] Majak, M., Zolnierek, A.: Rough sets approach to the problems of classification. In: *Proceedings of International Conference MOSIS X.*, pp. 109–114 (2010)
- [13] Zolnierek, A., Majak, M.: Rough sets approach to the classification task with modification of decision rules. In: *Proceedings of the 11th WSEAS International Conference on Systems Theory and Scientific Computation*, pp. 53–56 (2011)

MLG: Enhancing Multi-label Classification with Modularity-Based Label Grouping

Piotr Szymański^{1,2} and Tomasz Kajdanowicz¹

¹ Wrocław University of Technology, Wrocław, Poland

Faculty of Computer Science and Management, Institute of Informatics

² illimites Foundation, Wrocław, Poland

piotr.szymanski@illimites.edu.pl, tomasz.kajdanowicz@pwr.wroc.pl

Abstract. Multi-label classification on data sets with large number of labels is a practically viable and intractable problem. This paper presents an optimization method for the multi-label classification process for data with a high number of labels. The newly proposed method starts with label grouping using community detection methods on interconnectedness graph of labels based on support sizes for every pair of labels. The grouping process is based on modularity-oriented community detection methods. Next the data instances are classified separately for each label community and the resulting labellings are merged afterwards. Both theoretical analysis and experimental results are provided. Experimental results comparing common classification methods to proposed Modularity-based Label Grouping (MLG) with embedded Binary Relevance, executed on on differentiated data sets show a performance increase by 27-41% compared to standard binary relevance, by 72-81% compared to RAKel and by several dozens compared to ECOC-BR-BCH with none or negligible difference in classification quality.

Keywords: multi-label classification, modularity, label grouping, community detection, label co-occurrence, label interconnectedness.

1 Introduction

The multi-label classification is an intractable problem as it tries to capture the dependency that exist between class labels while finding the mapping function. In this paper we propose a method for optimizing performance of multi-label classification utilizing the information about label interconnectedness based on support sizes of every pair of labels. Large number of labels becomes a computational barrier for many classification algorithms, making non-linear classification methods infeasible. In order to improve multi-label classification or in many cases make such a process tractable, we introduce a method for dividing labels into coherent groups while trying to minimize the loss of information in the process. Such a division into label communities provides a possibility to label a dataset in each of the communities and treat the union of all sub-classifications as the final classification.

To facilitate this process we employ methods used in social networks to detect communities based on the notion of modularity. For a multi-label classification problem with labels $\lambda_i \in \mathcal{L}$ we construct a graph of label interconnectedness $G(\mathcal{L}, E)$. Every

pair of labels (λ_i, λ_j) is connected if and only iff there exists a positive support[1] for those two labels in the given data set, while the value of this support (which is the percentage of data instances that support those labels) become an edges weight. For such a graph we apply a method of community detection that approximates the division with maximum modularity. The concept behind modularity is that true community structure in a network corresponds to a statistically surprising arrangement of edges which can be quantified up to a multiplicative constant, by the number of edges falling within groups minus the expected number in an equivalent network with edges placed at random.[2].

Such communities can be interpreted as clusters of labels which share a relevant (distant from a random situation) portion of data, i.e. there exist a significant number of data instances that admit at least two labels from a given community while admitting very few or none labels from outside the community. Such a division should be a good approximation of the classification done on all of the labels, which generally depends on the number of instances supporting any produced classification.

Obviously the benefits of such optimization vary depending on the data and whether or with what quality do the data posses structure of interdependency. This is both relative to the quality of data and the validity of proposed labels understood as whether the labels correspond to actual eidetic phenomena and whether these phenomena are represented in the collected data. Fortunately community detection algorithms that maximize modularity allow a characterisation of the data also in terms of inability to divide data into communities, thus informing about bad quality data, bad label choice or high levels of randomness of underlying phenomena, which is not apriori accessible in current classification methods.

2 Multi-label Classification

The standard, conventional classification assumes each instance is associated with exactly one of a finite set of possible classes. An extended classification problem may allow instances to be associated with several labels simultaneously, which is addressed by multi-label classification, usually denoted as a label-set. Classical classification aims to learn a function f that maps an input $x \in X$ to an output class $c \in C$, i.e., results of classification—values y belong to only one of the classes from C , $X \rightarrow C$. Multi-class classification is a mapping from an input $x \in \mathcal{X}$ to an output $y \in \mathcal{Y}$ where Y denotes a set of classes with m number of classes $C = \{c_1, c_2, \dots, c_m\}$. Multi-label classification is a mapping from an input $x \in \mathcal{X}$ to an output $\Lambda \in 2^{\mathcal{L}}$, where label λ_i from all possible labels \mathcal{L} is a textual description of a class $c_i \in \mathcal{C}$.

2.1 Classification Algorithms

Binary Relevance. One of the most common problem transformation method in multi-label classification is *binary relevance (BR)*. This is due to its simplicity and, often surprising, high accuracy. *BR* learns l binary classifiers, one for each different label in \mathcal{L} . The initial multi-label data set is transformed into l single-label data sets, one D_{λ_j} for λ_j label, $j = 1, 2, \dots, l$. Each data set contains all instances of the original data set. The instances in D_j are labeled positively if the λ_j was is the label set of the original

instance and negatively otherwise. In the inference phase for new instance BR outputs the union of positively predicted labels by all l binary classifiers.

The computational complexity of learning in binary relevance method obviously depends on the size of label set $|\mathcal{L}| = l$. It might be quantified as $O(l \times f(d, N))$, where $f(d, N)$ denotes the learning complexity of underlying base classifier on data set with d attributes and N instances. The testing phase has the complexity $O(l \times g(d))$, where $g(d)$ denotes the inference complexity of base classifier.

Label Power-Set. Another problem transformation method is *label power-set (LP)*. The general idea of the method is to transform distinct combinations of labels into new classes and treat the problem as single-label one. Thus, in the method each distinct set of labels from the original training data set is mapped to a new, different class in a new single-label classification task. Given a new instance, the single-label classifier of LP outputs the most probable single class which can be easily translated into a set of labels.

However, the simplicity of the method does not go hand in hand with its complexity. The computational cost of learning in label power-set method is upper bounded by the learning complexity of $\min(N, 2^l)$ distinct label-sets. Therefore the complexity equals to training cost of single classification $O(\hat{f}(d, N))$, where \hat{f} needs to deal with $\min(N, 2^l)$ classes, which is usually much larger than l . The inference in LP multi-label classifier needs to consult only single model. Thus, its inference complexity equals to $O(\hat{g}(d))$, where $\hat{g}(d)$ denotes inference cost of \hat{g} multi-class base classifier of an instance with d attributes. Nonetheless, the label power-set method has very limited generalization abilities as it is not able to provide an output of unseen label-sets.

RAkEL. One of the very successful multi-label classification methods based on ensemble idea is *Random k-label-sets (RAkEL)* [3]. It constructs an ensemble of label power-set multi-label classifiers for randomly obtained subsets of labels. Therefore it randomly breaks a large set of labels into n subsets of size k . Obtained subsets are usually small and are called k-label-sets. Each of obtained data sets is then generalized using the LP method. Thus, after training the methods has n LP models. In the inference phase given a new instance it queries these models and averages their decisions per label. In order to obtain final decision a threshold function is needed.

RAkEL method provides a learning scheme that is computationally simpler than *label power-set*. It transforms initial multi-label problem into smaller sub-problems. The computational cost of learning equals to $O(n \times \hat{f}(d, N))$, where \hat{f} needs to deal with k classes and n is the number of classifiers. The inference phase requires to query n LP models. Thus, testing has the complexity of $O(n \times \hat{g}(d))$, where $\hat{g}(d)$ denotes the inference complexity of LP classifier.

RAkEL multi-label classification method is able to predict unseen label-sets. Nevertheless, RAkEL's accuracy strictly depends on the size of k in k-label-sets and a number n of trained models. In order to obtain final classification results voting is performed. There are $\frac{nk}{l}$ votes per label in average and the larger it is, the higher the effectiveness. However, parameters k and n can not be chosen randomly. The best behaviour of the method is for k small enough to deal with LP problems and n large enough to obtain more votes [3].

ECOC-BR-BCH. The ECOC-BR-BCH method is based on the multi-label classification framework that replaces the original description of multiple labels assigned to the trained instances using encoding technique [4]. Then, instead of learning the multi-label classifier on original labels, it tries to learn on the encoded ones. In the inference phase, the method results in encoded classification output y_m^{enc} and the final decision needs to be decoded into original multi-label space – y_m .

Hence, the framework provides a general method matching different coding and error correcting approaches ECOC with different multi-label classification methods it was utilized its version based on *BR* and *BCH* (Bose-Chaudhuri-Hocquenghem) code, which was reported as the best one [4].

2.2 Evaluation Metrics

In order to quantify the results there were applied some basic measures. The first one Hamming Loss *HL* is defined as:

$$HL = \frac{1}{N} \sum_{i=1}^N \frac{Y_i \Delta F(x_i)}{|Y_i|} \quad (1)$$

where: N is the total number of instances x in the test set; Y_i denotes actual (real) list of labels for instance x_i , $F(x_i)$ is a sequence of labels predicted by multi-label classifier for instance x_i and Δ stands for the symmetric difference of two vectors, which is the vector-theoretic equivalent of the exclusive disjunction in Boolean logic.

The second evaluation measure is Classification Accuracy *CA* [5], defined as:

$$CA = \frac{1}{N} \sum_{i=1}^N I(Y_i = F(x_i)) \quad (2)$$

where: N , Y_i , $F(x_i)$ have the same meaning as in Eq. 1, $I(true) = 1$ and $I(false) = 0$.

Measure *CA* provides very strict evaluation as it requires the predicted set of labels to be an exact match of the true set of labels.

The third evaluation measure - Accuracy *ACC* [6], is defined as:

$$ACC = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^l I(Y_{ij} = F(x_{ij})) \quad (3)$$

where: N denotes the number of data instances, Y_{ij} denotes the actual j th label of the i th data instance, and $F(x_{ij})$ is the j th label predicted by multi-label classifier for the i th data instance.

On the other hand, label-based evaluation measures, instead of being calculated separately for each instance, can be evaluated separately for each label and averaged across all labels. Label based evaluation measures can be divided into: macro-averaging and micro-averaging measures.

The former measures are calculated as an average measure obtained for each of labels separately. Using the notation where *tp* denotes *true-positive*, *fp* - *false-positive*,

tn true-negative and fn - false-negative [7] each macro averaging measure M_{macro} can be calculated as in Equation 4.

$$M_{macro} = \frac{1}{|\mathcal{L}|} \sum_{i=1}^{|\mathcal{L}|} M(tp_i, fp_i, tn_i, fn_i) \quad (4)$$

It is assumed that the particular measure $M(tp, fp, tn, fn)$ is a binary evaluation measure calculated on the contingency table. It means that macro-averaging measures ordinary average a binary measure.

On the other hand, the latter, micro-averaging measures, are calculated for all labels jointly. We can observe that in micro-averaging measures, labels are treated as different instances of the same global label. It is expressed by the summation for each of tp , fp , tn and fn counts, see Equation 5.

$$M_{micro} = M\left(\sum_{i=1}^{|\mathcal{L}|} tp_i, \sum_{i=1}^{|\mathcal{L}|} fp_i, \sum_{i=1}^{|\mathcal{L}|} tn_i, \sum_{i=1}^{|\mathcal{L}|} fn_i\right) \quad (5)$$

3 Community Structure Discovery

Community structure discovery is an interesting problem in network analysis. The aim of community discovery is to divide the set of vertices of a given network into disjoint subsets (communities) with regard to a given measure of differentiation. Community discovery is performed in relation to the structure of the network taking into account the distribution of edges between nodes, their directedness or weight. For example in the network of WWW pages with edges representing links between pages communities represent concentration on a similar topic[8] or in social networks they would correspond to forms of social organization such as social communities.

Upon applying community structure methods one usually assumes, that the network of interest divides naturally into subgroups and the experimenter's job is to find those groups. The number and size of the groups are thus determined by the network itself and not by the experimenter. Moreover, community structure methods may explicitly admit the possibility that no good division of the network exists, an outcome that is itself considered to be of interest for the light it sheds on the topology of the network [9].

For a graph $G(V, E)$ the output of a community detection algorithm is a partition $C(V)$ of the vertex set V . In terms of modularity-oriented or modularity-maximizing community detection, the subsets in the partition are constructed to maximize the modularity of the division defined as:

$$Q = \frac{1}{2|E|} \sum_{u,v \in V} \left(A_{uv} - \frac{deg(u)deg(v)}{2|E|} \right) \llbracket \exists (V_i \in C(V))(u, v \in V_i) \rrbracket$$

Where $\frac{deg(u)deg(v)}{2|E|}$ represents an average number of edges starting or ending in u or v in a Configuration Model graph following the same degree distribution as the original graph G . This graph is obtained by taking each edge, cutting it into two halves (called stubs in the model) and reattaching each of the halves to a randomly chosen node[10];

A_{uv} is the number of edges between u and v , $deg()$ is the degree of a given vertex, $|E|$ is the number of edges in graph G and $\mathbb{I}[\exists(V_i \in C(V))(u, v \in V_i)] = 1$ if vertices u and v are in the same community and $= 0$ otherwise. Many forms of equivalent statement of modularity are provided in literature, expressed in terms of eigenvalues[9] or in a matrix form[11]. If $Q \approx 0$ then the data is nearly random and does not admit a sensible community detection, such information obtained early saves time in doing senseless classification. If $Q > 0$ then the data may admit a community structure.

4 Proposed Method

We define the input data for an instance of multi-label classification problem as a set of labels $\mathcal{L} = \lambda_1, \dots, \lambda_n$ and a set of instances represented as a matrix T of rows which represent instances and columns which represent labels and $T_{ij} = 1$ when an instance i supports label j . We start by counting supports for every pair of labels, i.e. generating a set $S = \{(\lambda_i, \lambda_j, support(\lambda_i, \lambda_j) : \lambda_i, \lambda_j \in \mathcal{L}, i < j)\}$. Next we build a weighted undirected graph $G(\mathcal{L}, S)$ and apply the fast greedy community detection algorithm[12] to G to obtain a family of communities $C(G) = \{C_k \mid C_k \subset \mathcal{L}\}$. For each community C_k we construct the matrix T_k consisting of data instances from the original matrix T with columns representing labels that are not in C_k removed. We run the classification algorithm against new multi-label classification problems with a set of labels C_k and data instances matrix T_k obtaining a family of classifications $CL_{k,i}$ indexed by community number and number of instances. We construct the final classification CL_i for each instance using $CL_i = \bigcup_{a=1}^m CL_{a,i}$.

The division approximating a maximum modularity Q can be obtained using a variant of other methods such a greedy algorithm[12] ($O(|\mathcal{L}| \log^2 |\mathcal{L}|)$), eigenvalue calculation[9] ($O(|E| |\mathcal{L}| + |\mathcal{L}|^2)$) or by other methods like simulated annealing as the problem is equivalent to a Potts spin glass problem[13].

Algorithm 1. Pseudocode for the proposed method

```

for all  $\lambda_i, \lambda_j \in \mathcal{L}^2$  do
  weight  $\leftarrow$  CALCULATE – SUPPORT – SIZE(trainingData,  $\lambda_i, \lambda_j$ )
  if weight  $\geq$  threshold then
    edges.append( $\lambda_i, \lambda_j, weight$ )
  end if
end for
C  $\leftarrow$  GREEDY – DIVIDE – INTO – COMMUNITIES(GRAPH(L, edges))
labels  $\leftarrow$   $\emptyset$ 
for a := 1 to SIZE(C) do
  labels.append(MULTI-LABEL-CLASSIFY(trainingData, dataInstances, C[a]))
end for
return labels

```

5 Experimental Results

5.1 Implementation

Experiments were carried in R and Matlab. The R-implemented part was used for data preprocessing, counting a normalized support size for every pair of data instances with a 100% meaning that all instances are labelled with both labels. The eclat algorithm implementation from the arules package was used[14] to calculate support sizes, the igraph library was used for community detection using the fast greedy algorithm[12]. Multi-label classification was executed in Matlab using perClass classification toolbox as a implementation for the base classifiers. There were implemented common multi-label classifiers: BR - Binary Relevance, LP - Label Powerset, ECOC-BR-BCH - Binary Relevance based on error correcting output code BCH, Rakel and the proposed method MLG-BR. As a base classifier the random forest with 200 trees was utilized.

5.2 Data Sets

The *medical*[15] dataset is based on the Computational Medicine Center’s 2007 Natural Medical Natural Language Processing Challenge and contains clinical free text reports labelled with disease codes. Another dataset, *enron*[16], is based on annotated email messages exchanged between Enron Corporation employees. The last dataset, *genbase*[17], refers to protein classification.

The basic statistics of datasets used in experiments, such as the number of data instances, the number of attributes, the number of labels, labels’ cardinality, density and distinct number of label-sets are presented in Table 1. It is assumed that label’s cardinality denotes the average number of labels per data instance, density is a fraction of average number of labels per instance to number of distinct labels and distinct label-sets indicate number of distinct combinations of labels in the observed instances.

Table 1. Data sets used in the experiments

Data	Instances	Attr.	Labels	Card.	Density	Distinct	Domain
genbase	662	1186	27	1.242	0.046	32	biology
medical	978	1449	45	1.245	0.028	94	text
enron	1702	1001	53	3.378	0.064	753	text

5.3 Results

We construct the description of the results for each data set as follows: the obtained community division and its modularity is presented, followed by the scores in different accuracy measures obtained by used methods. Those scores are pictured per measure in Fig. 1 for every method compared per data set. We conclude every result with a performance comparison.

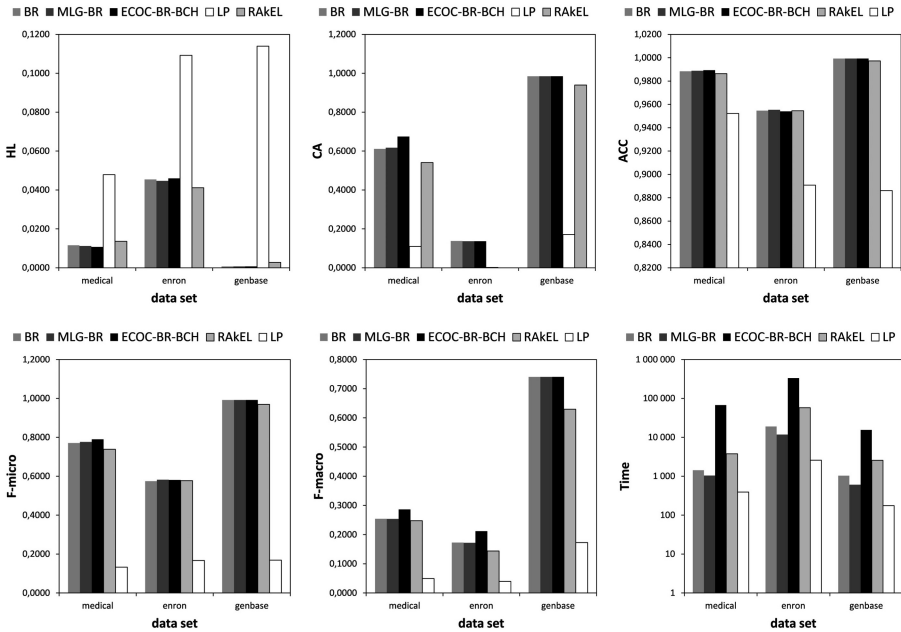


Fig. 1. Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results per method per data set

The *genbase* data set admits a modularity of $Q = 0.660474$ achieved by a division into a family of 20 communities consisting of 16 singletons, 2 sets of 2, 1 set of 3 and 1 set of 4 labels. For this data set MLG-BR delivered classification of identical quality with no difference in CA, ACC, fmicro and fmacro with respect to BR or ECOC-BR-BCH. At the same time MLG-BR was faster by 41% than BR and by more than 25 times than ECOC-BR-BCH.

The 45-label data set *medical* was divided into 43 communities with two of them containing 2 labels and other being singletons achieving a modularity $Q = 0.2498513$. Comparing to BR, MLG-BR provided performance improvement by 27% with CA, ACC, fmicro and fmacro practically identical suffering only from an increase of Hamming Loss by 3%. MLG-BR provided huge performance improvement (60 times) in comparison to ECOC-BR-BCH.

Enron was the largest data set used for experimentation. This set of 53 labels was divided in 25 sets with 23 singletons and two bigger communities with 16 and 14 labels respectively achieving a modularity $Q = 0.1135487$. In this case MLG-BR was actually better than BR, with CA increase by 1% and HL decrease by 2% with no changes in ACC, fmacro and a decrease by 1% in fmicro. It was also better than ECOC-BR-BCH with a decrease by 3% in Hamming Loss, identical CA, ACC and fmicro measures with a decrease by 19% in fmacro. Such excellent results were obtained using MLG-BR that was faster by 38% than BR and more than 30 times than ECOC-BR-BCH.

6 Conclusions and Future Work

This paper presented an optimization method for the multi-label classification process for data with a high number of labels. Label grouping based on interconnectedness and modularity-oriented community detection turned out to be an effective method of label-space decomposition. We conclude that MLG-BR provided an excellent performance improvement (up to one order of magnitude in processing time) with none or negligible differences in CA and HL over BR and ECOC-BR-BCH for data sets admitting a label grouping that achieves relevant modularity. For data sets with a more random structure of label co-occurrence relationships between labels MLG-BR still provides a significant increase in speed with a moderate loss in CA and increase in HL. In general MLG-BR performs better quality classification than Rakel or LP just like BR and ECOC-BR-BCH in terms of CA or HL at the same time outperforming Rakel by 72-81%.

In the future we would like to provide a detailed theoretical analysis of the method and the relation between performance improvement, classification quality and characteristics of the data such as instance/label count, label distributions or modularity achieved by community detection in the label interconnectedness graph etc. Another interesting direction is the analysis of the impact of resolution limit of modularity-based community detection[18] on the modularity-based label grouping multi-label classification. Another task is to execute experimental data on very large sets of data and apply MLG optimization to other classification methods like Rakel or LP. We would also like to publish extended results we have already calculated which we decided to remove from this paper to fulfill page number requirements.

References

- [1] Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. *ACM SIGMOD Record* 22(2), 207–216 (1993)
- [2] Newman, M., Girvan, M.: Finding and evaluating community structure in networks. *Physical Review E* 69(2), 026113 (2004)
- [3] Tsoumakas, G., Vlahavas, I.P.: Random k -labelsets: An ensemble method for multilabel classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) *ECML 2007*. LNCS (LNAI), vol. 4701, pp. 406–417. Springer, Heidelberg (2007)
- [4] Kajdanowicz, T., Kazienko, P.: Multi-label classification using error correcting output codes. *International Journal of Applied Mathematics and Computer Science* 22(4), 829–840 (2012)
- [5] Ghamrawi, N., McCallum, A.: Collective multi-label classification. In: *Proceedings of International Conference on Information and Knowledge Management*, pp. 195–200. ACM (2005)
- [6] Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Dai, H., Srikant, R., Zhang, C. (eds.) *PAKDD 2004*. LNCS (LNAI), vol. 3056, pp. 22–30. Springer, Heidelberg (2004)
- [7] Bishop, C.M.: *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer-Verlag New York, Inc., Secaucus (2006)

- [8] Gibson, D., Kleinberg, J., Raghavan, P.: Inferring Web communities from link topology. In: Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia: Links, Objects, Time and Space—Structure in Hypermedia Systems Links, Objects, Time and Space—Structure in Hypermedia Systems - HYPERTEXT 1998, pp. 225–234. ACM Press, New York (1998)
- [9] Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences of the United States of America 103(23), 8577–8582 (2006)
- [10] Hofstad, R.V.D.: Random Graphs and Complex Networks (2013), <http://www.win.tue.nl/~rhofstad/NotesRGCN.pdf> (accessed April 30, 2008)
- [11] Newman, M.E.J.: Detecting community structure in networks. The European Physical Journal B - Condensed Matter 38(2), 321–330 (2004)
- [12] Newman, M.: Fast algorithm for detecting community structure in networks. Physical Review E 69(6), 066133 (2004)
- [13] Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. Physical Review E 74(1), 016110 (2006)
- [14] Michael Hahsler, B.G.: Introduction to arules: Mining Association Rules and Frequent Item Sets
- [15] Pestian, J.P., Brew, C., Matykiewicz, P., Hovermale, D.J., Johnson, N., Cohen, K.B., Duch, W.: A shared task involving multi-label classification of clinical free text. In: Proceedings of the Workshop on BioNLP 2007 Biological Translational and Clinical Language Processing BioNLP 2007, vol. 1, pp. 97–104 (2007)
- [16] Klimt, B., Yang, Y.: The Enron Corpus: A New Dataset for Email Classification Research. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 217–226. Springer, Heidelberg (2004)
- [17] Diplaris, S., Tsoumakas, G., Mitkas, P.A., Vlahavas, I.: Protein Classification with Multiple Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 448–456 (2005)
- [18] Kumpula, J.M., Saramäki, J., Kaski, K., Kertész, J.: Limited resolution in complex network community detection with Potts model approach. The European Physical Journal B 56(1), 41–45 (2007)

Intelligent System for Channel Prediction in the MIMO-OFDM Wireless Communications Using a Multidimensional Recurrent LS-SVM

Jerzy Martyna

Institute of Computer Science, Faculty of Mathematics and Computer Science
Jagiellonian University, ul. Prof. S. Łojasiewicza 6, 30-348 Cracow, Poland

Abstract. In order to resolve channel prediction in the multiple-input multiple-output orthogonal frequency division multiple (MIMO-OFDM) system used in wireless communication, a novel intelligent system based on least squares support vector machines (LS-SVMs) is proposed in this paper. To manipulate the iterative problem, the recurrent multidimensional version LS-SVM has been used. The proposed algorithm used in this system allows us to implement nonlinear decision regions in the channel prediction in MIMO-OFDM systems, and adaptively convergent to minimum mean squared error solutions. It is shown by simulation that the proposed method is able to provide accurate results in channel prediction in these systems. Moreover, this method can be also used in many signal degradations caused by multipath propagation, shadowing from obstacles, etc.

Keywords: wireless channel estimation, multidimensional recurrent least-squares support vector machine, multiple-input multiple-output (MIMO) channel model, OFDM system.

1 Introduction

Hybrid computational intelligence is defined as any effective combination of intelligent techniques that performs superior or, in a competitive way to simple standard intelligent techniques. Hybrid intelligence was in fact attempted in several papers, for instance see [2], [5], as an extension to the standard experimentation with most of the well-known intelligent techniques, in various application domains.

Recently, support vector machines (SVMs) have been introduced by Vapnik [20], [19] and his colleagues [4] as a new method for solving classification and static function approximation problems. The classical training algorithm for SVMs is equivalent to solving a quadratic programming with linear and inequality constraints. Least squares support vector machines (LS-SVMs) are introduced by Suykens et al. [17], [16] as a reformulation of standard SVMs. LS-SVMs simplified the training process for a standard SVM by replacing the inequality constraints with equality ones, Suykens and Vandewalle [18] proposed

the recurrent LS-SVM, in which a class of nonlinear output error models for the autonomous case has been investigated.

In this paper, a multidimensional recurrent version of the LS-SVM is proposed as an intelligent system for solving the channel prediction problem in wireless communication based on multiple-input multiple-output (MIMO) techniques. The orthogonal frequency division multiple access (OFDMA), also known as multiuser OFDM, was coupled with multiple-input multiple-output (MIMO) techniques for reliable and high speed wireless communication over frequency-selective radio channels. Multiple antenna systems are one of the key technologies for the next generation wireless communications based on the MIMO-OFDM systems, especially in WCDMA-based 3G systems [1], Mobile WiMAX, LTE, IEEE 802.11a/n/ac, wireless PAN (MB-OFDM) [12], broadcasting (DAB, DVB, DMB), etc.

This MIMO-OFDM technique has attracted much attention due to its advantage in capacity as well as the ability to support multiple users simultaneously [9]. It is based on the assumption that the receiver and transmitter have knowledge of the channel coefficients. In reality they must either be estimated or predicted. Some popular ways to estimate the channel coefficients are by using *pilot symbols* [8]. Nevertheless, this method wastes time learning the channel parameters when meaningful data can be sent.

In some works, a feedback or a partial feedback is used to transmission of the so-called *channel state information* (CSI) from the receiver to the transmitter [10]. To have a good CSI information some methods to estimate of the MIMO-OFDM channel [7], [3] were developed. However, these methods waste time while learning the channel when meaningful data can be sent.

A number of channel estimation methods have already been proposed for MIMO-OFDM systems. When the full or partial information of the channel correlation is known, a good channel performance can be achieved via some *minimum mean square error* (MMSE) methods [11]. By using decision feedback symbols, the Takagi-Sugeno-Kang (TSK) fuzzy approach proposed by Zhang et al. [21] can achieve a performance similar to the MMSE methods while with a low complexity. Channel estimation for OFDM systems using adaptive Radial Basis Function (RBF) networks has been proposed by Zhou et al. [23].

The use of the SVM method and recently developed the LS-SVM method have been proposed to solve a number of digital communication problems. Among others, signal equalization and detection for a multicarrier (MC)-CDMA system based on SVM linear classification has been investigated by Rahman et al. [13]. In the paper by Sánchez-Fernández et al. [14] SVM techniques have been used for a nonlinear channel estimation for multiple-input multiple-output systems. The SVM technique for a robust channel estimation in the OFDM data structure was proposed by Fernández-Getino Garcia et al. in the paper [6]. However, none of these solutions can be used easily in on-line adaptive algorithms.

The goal of this paper is to create a multidimensional recurrent LS-SVM method for channel prediction in wireless communication. There are two innovations in this proposal. It used a multidimensional recurrent LS-SVM for

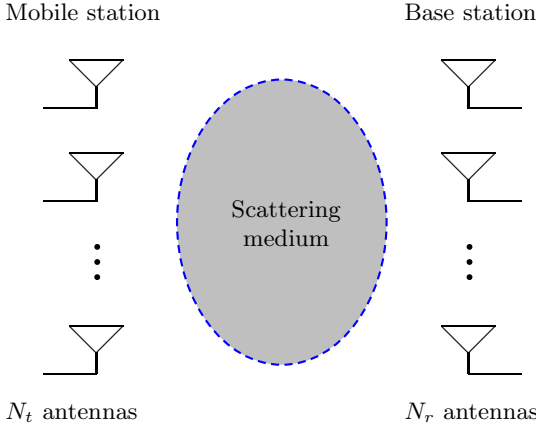


Fig. 1. Two antenna arrays in a scattering environment

channel prediction in a MIMO-OFDM system in which inequality constraints are replaced by equality constraints and a quadratic error criterion was used. Secondly, the adequacy of the beam-forming was shown to be ideal for multidimensional recurrent LS-SVM predicted MIMO-OFDM channels.

The rest of the paper is organized as follows. In the next section we formulate the MIMO-OFDM received model. In section 3 we present our solution based on the multidimensional recurrent LS-SVM approach. The results of the simulation of the received SNR in the MIMO-OFDM system is presented in section 5. In section 6 we give our concluding remarks.

2 The Theoretical Background

In this section we present an overview of one of simplest statistical channel model for MIMO-OFDM system.

We assume the existence of the un-coded and beam-forming MIMO-OFDM system. For both two cases are considered: the first, in which the transmitter and the receiver have a full channel state information (CSI), and the second, when both the transmitter and the receiver have the prediction matrix.

2.1 The Un-Coded MIMO-OFDM System

Consider a MIMO-OFDM system of N_r receiver antennas and N_t transmitter antennas as illustrated in Fig. 1.

A narrowband MIMO-OFDM channel \mathbf{H} can be statistically expressed with an $N_r \times N_t$ matrix as

$$\mathbf{H} = \Theta_R^{1/2} \mathbf{A}_{iid} \Theta_T^{1/2} \tag{1}$$

where Θ_R and Θ_T are correlation matrices for the receiver antennas and transmitter antennas, respectively, while \mathbf{A}_{iid} represents an i.i.d (independent and identically distributed) Rayleigh fading channel. The basic assumption behind the correlation matrix-based MIMO channel model in Eq. (1) is that the correlation matrices for the transmitter and the receiver can be separated. That particular assumption holds when the antenna spacing in the transmitter and the receiver is sufficiently smaller than the distance between the transmitter and the receiver, which usually is true for most of wireless communication environments.

Let $y_m(t)$ denote the received signal at the received antenna. Then, the received signals at the received antenna are denoted as $\mathbf{y}(t) = [y_1, y_2(t), \dots, y_{N_r}(t)]^T$. Similarly, the transmitted signals at the transmitter antenna are denoted as $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_{N_t}(t)]^T$, where $x_n(t)$ is the signal transmitted at the n antenna element. The relation between the transmitter antennas and the receiver antennas signals can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (2)$$

where \mathbf{y} is the $N_r \times 1$ received vector, \mathbf{x} is the $N_t \times 1$ transmitted symbol vector with each x_i belonging to constellation \mathcal{C} with symbol energy E_s , and \mathbf{n} is the while noise vector of size $N_r \times 1$ with $n_i \sim \mathcal{CN}(0, N_0)$. The channel state matrix $\mathbf{H} = \{h_{mn}\}$ gives a complex channel gain between the m -th receiver and the n -th transmit antenna.

2.2 The Beam-Forming MIMO System

In the beam-forming MIMO system, the received symbols are expressed in two scenarios, when the transmitter and the receiver have full channel information (CSI) and when they have the prediction matrix. Then, the channel matrix $\mathbf{H} = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}^H$ is a singular value decomposition (SVD) where \mathbf{U} and \mathbf{V} are unitary matrices corresponding to the i -th non-zero singular value $\sigma_H(i)$, ($\sigma_H(1) \leq \dots \leq \sigma_H(M)$) and $M = \text{rank}(\mathbf{H})$. Assuming that $\tilde{x} = v_1 \cdot \mathbf{x}$ we can obtain from Eq. (1) the received symbols u_1^H as follows:

$$u_1^H \mathbf{y} = \sigma_H(1)x + u_1^H \mathbf{n} \quad (3)$$

Assuming $\tilde{u} = \mathbf{u}_1^H \mathbf{n}$ we can obtain

$$E |\tilde{u}|^2 = N_r \cdot N_0 \quad (4)$$

Thus, the channel within a MIMO system is time-varying and can be expressed in a matrix notation as $\mathbf{H} = \hat{H} + E$. Therefore, the received symbols are as follows:

$$\hat{u}_1^H \mathbf{y} = (\sigma_H(1) + \hat{u}_1^H E \hat{v}_1) \mathbf{x} + u_1^H \mathbf{n} \quad (5)$$

In the general case the Doppler spread of the signal is greater than the pulse bandwidth. Assuming a coded beamforming scheme for frequency flat MIMO

fading channels, the channel coefficient of the m -th receiver and the n -th transmit antenna can be formulated as follows [22]:

$$h_{mn}(t) = h_{mn}(k) + jh_{mn}(k) \quad (6)$$

where in-phase component is represent as

$$h_{mn}^I(k) = \sqrt{\frac{2}{M}} \sum_{n=1}^M \cos(2\pi f_d k \sin(\alpha_n) + \Phi_n) \quad (7)$$

and the quadrature component can be written as

$$h_{mn}^Q(k) = \sqrt{\frac{2}{M}} \sum_{n=1}^M \cos(2\pi f_d k \sin(\alpha_n) + \Psi_n) \quad (8)$$

where $\alpha_n = \frac{2\pi n - \pi + \Theta}{4M}$ and Φ_n, Ψ_n, Θ are $U[-\pi, \pi]$.

3 The Multidimensional Recurrent LS-SVM

The recurrent LS-SVM based on the sum squared error (SSE) to deal with the function approximation and prediction has been proposed by Suykens and Vandewalle [18]. However, the so defined recurrent LS-SVM will not be adequate for the channel prediction in the MIMO system. It is caused by the lack of the high-dimensional reconstructed embedding phase space.

In order to extend the recurrent least squares vector machine to a multidimensional recurrent LS-SVM we introduce scalar time series $\{s_1, s_2, \dots, s_T\}$ in the form

$$\hat{s}_k = f(\hat{s}_{k-1}), \quad k = m', m' + 1, \dots, N' + m' - 1 \quad (9)$$

where m', N' are referred to the embedding dimension and the number of training data, respectively. The function approximation is given by

$$\hat{s}_k = \mathbf{w}^T \phi_i(\hat{s}_{k-1}) + b, \quad k = m', m' + 1, \dots, N' + m' - 1, \quad i = 1, 2, \dots, m' \quad (10)$$

where $\mathbf{w} = [w_1, w_2, \dots, w_{m'}]$ is the output weight vector, $b \in R$ is the bias term $\phi(\cdot)$ is the nonlinear mapping function estimated by means of using training data. Thus, the recurrent LS-SVM can be formulated as the quadratic optimization problem:

$$\min_{w_i, b_i, e_{k,i}} \mathcal{J}(w_i, b_i, e_{k,i}) = \frac{1}{2} \sum_{i=1}^{m'} w_i^T w_i + \frac{\gamma}{2} \sum_{k=m'+1}^{N'+m'-1} \sum_{i=1}^{m'} e_{k,i}^2 \quad (11)$$

subject to the following constraints:

$$\begin{cases} s_k^{(1)} - e_{k,1} = w_1^T \phi_1(s_{k-1} - e_{k-1}) + b_1 \\ s_k^{(2)} - e_{k,2} = w_2^T \phi_2(s_{k-1} - e_{k-1}) + b_2 \\ \vdots \\ s_k^{m'} - e_{k,m'} = w_{m'}^T \phi_l(s_{k-1} - e_{k-1}) + b_{m'} \end{cases} \quad (12)$$

where $k = m' + 1, m' = 2, \dots, N' + m' - 1$, $e_k = s_k - \hat{s}_k$. Generally, the error term here is defined as

$$e_k = x_k - f(x_{k-1}) \tag{13}$$

The corresponding Lagrangian for Eq. (11) is given by

$$\begin{aligned} \mathcal{L}(w_i, b_i, e_{k,i}; \alpha_{k,i} = \mathcal{J}(w_i, b_i, e_{k,i}) \\ + \sum_{k=m'+1}^{N'+m'-1} \sum_{i=1}^{m'} \alpha_{k,i} [s_k^{(i)} - e_{k,i} - w_i^T \phi_i \times (s_{k-1} - e_{k-1}) - b_i] \end{aligned} \tag{14}$$

with respect to w_i, b_i and e_i . The solution given by the Karush-Kuhn-Tucker theorem is given by:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w_i} = w_i - \sum_{k=m'+1}^{N'+m'-1} \alpha_{k,i} \phi_i (s_{k-1} - e_{k-1}) = 0 \\ \frac{\partial \mathcal{L}}{\partial b_i} = \sum_{k=m'+1}^{N'+m'-1} \alpha_{k,i} = 0 \\ \frac{\partial \mathcal{L}}{\partial e_{k,i}} = \gamma e_k - \alpha_{k,i} - \sum_{i=1}^{m'} \alpha_{k+i,i} \frac{\partial}{\partial e_{k+m'-i,i}} \\ \quad \times [w_i^T \phi_i (s_{k-1} - e_{k-1})] = 0 \\ \frac{\partial \mathcal{L}}{\partial \alpha_{k,i}} = s_k^{(i)} - e_{k,i} - w_i^T \phi_i (s_{k-1} - e_{k-1}) - b_i = 0 \end{cases} \tag{15}$$

where $k = m' + 1, m' + 2, \dots, N' + m' - 1$ and $i = 1, 2, \dots, m'$.

Due to the application of the Mercer's condition [15] there exists a mapping and the LS-SVM model for the given problem, namely

$$\hat{s}_k = \sum_{i=m'+1}^{N'+m'-1} \sum_{p=1}^{m'} \alpha_{i,p} K_p(z_i, \hat{s}_{k-1}) + b_p \tag{16}$$

where $z_l = s_l - e_l$. The initial condition is given by $\hat{s}_i = s_i$ for $i = 1, 2, \dots, m'$. Thus, the kernel function $K_p(\cdot, \cdot)$ can be stated as follows:

$$K_p(x_i, x_j) = \phi_p^T(x_i) \phi_p(x_j) = \exp \left(- \frac{\|x_i - x_j\|^2}{2\sigma_p^2} \right) \tag{17}$$

where $p = 1, 2, \dots, m'$.

4 Experimental Results

We considered a multiuser MIMO-OFDM system model with an intelligent system and four transmit/receive antennas (see Fig. 2). The number of subcarriers was set to 512. Kernel width parameters σ and γ are selected using fivefold stratified cross validation on the training data set. Thus, 0.5 and 500 were chosen for σ and γ by trying in the range [0.1 - 5] and [0.1 - 1000], respectively. The LS-SVM was trained with a number of the training data for each MIMO sub-channel.

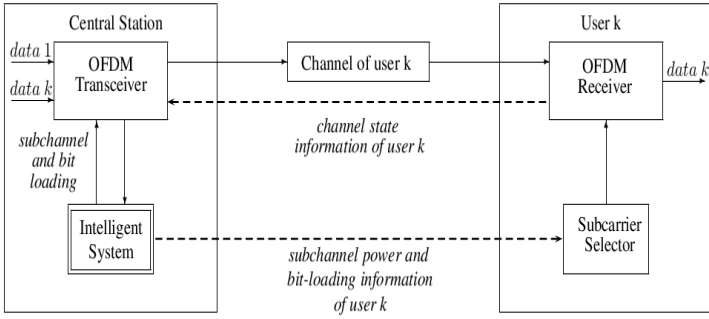


Fig. 2. Multiuser MIMO-OFDM system model

For each prediction, $N_p = 9$ and the previous 10 channel coefficients and their respective predictions were computed the *root mean square error* (RMSE) values, namely

$$RMSE(k) = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (h_{mn} - \hat{h}_{mn})^2} \tag{18}$$

In the first experiment a Rayleigh flat fast 4×4 MIMO channel with $f_d \cdot T_s = 0.05$ has been generated. The sub-channels are uncorrelated. To save space and redundancy the in-phase component of $h_{11}(k)$ was only considered. Fig. 3 shows the comparison of the actual and predicted values of amplitude in dependence on the sample number. In order to test the performance of the MIMO channel prediction, we used the received *signal-to-noise ratio* (SNR) in the general form

$$\rho = \frac{\sigma_x^2}{\sigma_e^2 + \sigma_n^2} \tag{19}$$

where σ_x^2 is the average received signal power, σ_e^2 is the predictive error, σ_n^2 is the average noise variance. Thus, after some algebraic manipulations for the un-coded system we can obtain

$$\rho_{uc} \triangleq \frac{E \|\hat{\mathbf{H}}\mathbf{x}\|_2^2}{E \|\mathbf{E}\mathbf{x}\|_2^2 + E \|\mathbf{n}\|_2^2} \tag{20}$$

and after several manipulations

$$\rho_{uc} = \frac{\sum_{i=1}^M E[\sigma_{\mathbf{H}}^2(i)]}{\sum_{i=1}^N E[\sigma_{\mathbf{E}}^2(i)] + \frac{N_r N_0}{E_s}} \tag{21}$$

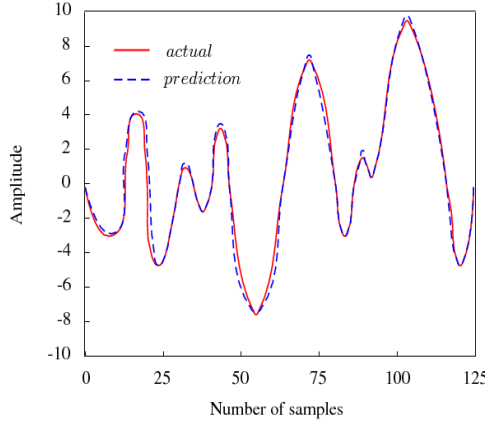


Fig. 3. Amplitude comparison for various number of samples

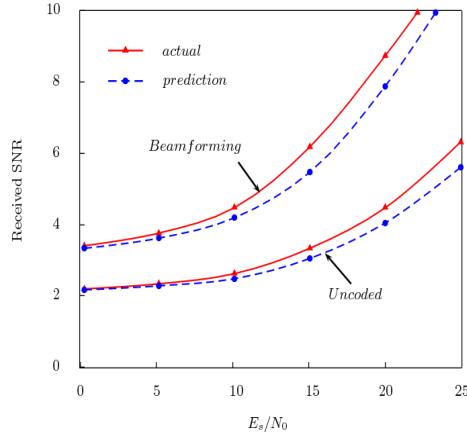


Fig. 4. Received SNR as a function of E_s/E_0 for un-coded and beam-forming MIMO system

where $N = rank(\mathbf{E})$, $\sigma_{\hat{\mathbf{H}}}(i)$ and $\sigma_{\mathbf{E}}(i)$ are the i -th non-zero singular values of $\hat{\mathbf{H}}$ and \mathbf{E} , respectively. The MIMO beam-forming can be formulated as follows

$$\hat{\mathbf{u}}_1^H \mathbf{y} = (\hat{\sigma}_1^2 + \hat{\mathbf{u}}_1^H \mathbf{x} + \tilde{n} \tag{22}$$

Thus, we can state the received SNR for the MIMO beam-forming system

$$\rho_{bf} = \frac{E[\sigma_1^2]}{E | \hat{\mathbf{u}}_1^H \mathbf{U} \mathbf{D} \mathbf{V}^H \hat{\mathbf{v}}_1 - \hat{\sigma}_{max} |^2 + \frac{N_r N_0}{E_s}} \tag{23}$$

Thus, comparing the above equation with the Eq. (19) we get the value of the prediction error σ_e^2 for the beam-forming prediction, namely

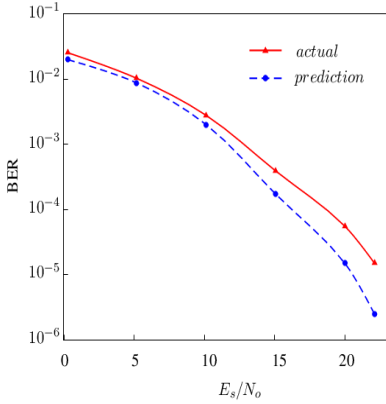


Fig. 5. Bit error rate (BER) as a function of E_s/E_0 for an un-coded 4×4 MIMO system.

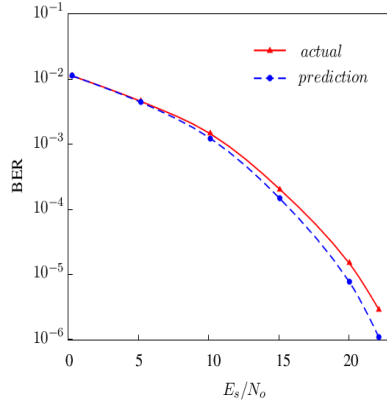


Fig. 6. Bit error rate (BER) as a function of E_s/E_0 for a beam-forming 4×4 MIMO system.

$$\sigma_e^2 = E \|\mathbf{E}\mathbf{x}\|_2^2 = E \|\hat{\mathbf{U}}^H \mathbf{E} \hat{\mathbf{V}}\mathbf{x}\|_2^2 \tag{24}$$

or after some algebraic manipulations

$$\sigma_e^2 = E |(\hat{\mathbf{u}}_1^H \mathbf{U} \mathbf{D} \mathbf{V}^H \hat{\mathbf{v}}_1 - \hat{\sigma}_{max})x|^2 \tag{25}$$

The behaviour of the received SNR for the 10000 binary phase shift keying (BPSK) symbol vectors with $E_s = 1$, $N_t = N_r = 2$ is given in Fig. 4. These graphs are obtained through a simulation for the typical value of ratio E_s/N_0 . This figure indicates that for a smaller value of E_s/N_0 the predicted value will be better adjusted than for a greater value of E_s/N_0 . The obtained values of the maximum singular value of the error matrix and the minimum singular value of the error matrix are given in Fig. 5 and Fig. 6, respectively.

5 Conclusion

A recurrent LS-SVM method was used in order to predict a MIMO channel. The received SNR for an un-coded and beam-forming MIMO system was derived. The proposed solution does not need to use the analytic mathematical model of performance measures. However, it can provide the parameters of the system in the varying time horizon. The experiment results show that the proposed method achieves the best performance for the RBF kernel function. This kernel function demands the use of scaling factors for all input parameters.

References

1. 3GPP: Tr25.869 tx diversity solutions for multiple antennas. v1.2.0 (2003)
2. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neuro-computing 72, 2729–2730 (2009)

3. Biguesh, M., Gershman, A.: Training-based mimo channel estimation: A study of estimator tradeoffs and optimal training signals. *IEEE Trans. on Signal Processing* 54(3), 884–893 (2006)
4. Cortes, C., Vapnik, V.: Support vector networks. *Support vector networks. Machine Learning* 20, 273–297 (1995)
5. Corchado, E., Suasel, V., Sedano, J., Hassanien, A.E., Calvo-Rolle, J.L., Slezak, D.: Soft computing models in industrial and environmental applications. In: 6th Int. Conf. SOCO 2011 (2011)
6. Garcia, M.F.G., Rojo-Álvarez, J., Alonso-Atienzo, F., Martínez-Ramón, M.: Support vector machines for robust channel estimation in ofdm. *IEEE Signal Processing Letters* 13(7), 397–400 (2006)
7. Ghogho, M., Swami, A.: Training design for multipath channel and frequency-offset estimation in mimo systems. *IEEE Trans. on Signal Processing* 54(6), 3957–3965 (2006)
8. Hao, X., Chizhik, D., Huang, H., Valenzuela, R.: A generalized space-time multiple input multiple output (mimo) channel model. *IEEE Trans. on Wireless Comm.* 3, 966–975 (2004)
9. Jindal, N., Vishwanath, S., Goldsmith, A.: On the duality of gaussian multiple-access and broadcast channels. *IEEE Trans. on Inf. Theory* 50(5), 768–783 (2004)
10. Shahtalebi, K., Bakhshi, G.R., Rad, H.: Full mimo channel estimation using a simple adaptive partial feedback method. *arXiv.org* (2007)
11. Ma, X., Yang, L., Giannakis, G.: Optimal training for mimo frequency-selective fading channels. *IEEE Trans. on Wireless Comm.* 4(2), 453–466 (2005)
12. van Nee, R., Prasad, R.: *OFDM for Wireless Multimedia Communications*. Artech House Publishers, Boston (2000)
13. Rahman, S., Saito, M., Okada, M., Yamamoto, H.: An mc-cdma signal equalization and detection scheme based on support vector machines. In: *Proc. 1st Int. Symp. Wireless Communication Systems*, pp. 11–15 (2004)
14. Sánchez-Fernández, M., de Prado-Cumlido, M., Arenas-García, J., Pérez-Cruz, F.: Svm multiregression for nonlinear channel estimation in multiple-input multiple-output systems. *IEEE Trans. on Signal Proc.* 52(8), 2298–2307 (2004)
15. Suykens, J., Gestel, T.V., Brabantter, J.D., Moor, B.D., Vandewalle, J.: *Least Squares Support Vector Machines*. World Sci. Pub. Co., Singapore (2002)
16. Suykens, J.A.K., Lukas, L., Vandewalle, J.: Sparse approximation using least squares support vector machines. In: *Proc. of the IEEE Int. Symp. on Circuits and Systems (ISCAS 2000)*, vol. 2, pp. 757–760 (2000)
17. Suykens, J.A.K., Vandewalle, J.: Least squares support vector machine classifier. *Neural Processing Letters* 9(3), 293–300 (1999)
18. Suykens, J.A.K., Vandewalle, J.: Recurrent least squares support vector machines. *IEEE Trans. Circuits Systems - I: Fundamental Theory and Applications* 47(7), 1109–1114 (2000)
19. Vapnik, V.: *Statistical Learning Theory*. John Wiley and Sons, New York (1998)
20. Vapnik, V.N.: *The Nature of Statistical Learning Theory*. Springer, Berlin (1995)
21. Zhang, J., He, Z., Wang, X., Huang, Y.: Tsk fuzzy approach to channel estimation for mimo-ofdm systems. *Signal Processing Letters* 14(6), 381–384 (2007)
22. Zheng, Y., Xiao, C.: Improved models for the generation of multiple uncorrelated rayleigh fading waveforms. *IEEE Trans. Commun. Letters* 6(6), 256–258 (2002)
23. Zhou, X., Wang, X.: Channel estimation for ofdm systems using adaptive radial basis function networks. *IEEE Trans. on Veh. Tech.* 52(1), 48–59 (2003)

Template-Based Synthesis of Plan Execution Monitors

Thomas Reinbacher¹ and César Guzmán-Alvarez²

¹ Embedded Computing Systems Group, Vienna University of Technology, Austria
`treinbacher@ecs.tuwien.ac.at`

² Universidad Politecnica de Valencia, Spain
`cguzman@dsic.upv.es`

Abstract. In the robotics domain, the state of the world may change in unexpected ways during execution of a task. From a planning perspective, these discrepancies may render the currently executed plan invalid and thus need to be detected as soon as possible. We tackle this problem by translating the problem of plan execution monitoring to a runtime verification problem. We propose a template based framework that allows detecting changes of the state during both plan generation and plan execution. We integrated our approach into a domain-independent platform for planning, executing, and monitoring.

1 Introduction and Related Works

During plan execution, discrepancies between the expected and the actual state of the world can stem from various sources. Rigorous execution monitoring is thus required to i) detect errors, ii) communicate with a replanning module or iii) to collect training instances for a learning module.

Plan execution monitoring needs to deal with the problem of monitoring actions in terms of its preconditions and effects, which typically represents an abstraction of the real problem. Several systems [10,12,3] monitor the continued validity of a plan during execution using annotations (variables to be monitored) of the plan. However, these annotations are generated without considering the goals of the planning problem. The work of [11] improved this process by including the goals of the planning problem through a regression algorithm. We build on these ideas from annotation-based approaches but separate the annotations from the plan to use them to instantiate our generic templates. Systems described by [9,14] perform plan execution monitoring only prior to the execution of an action but not during it's execution.

A technique with similar aims as plan execution monitoring is runtime verification [5] which has been successfully applied to a number of high-level programming languages. The authors of [15] report on an execution-monitoring approach with temporal logic. However, their specifications are defined as domain-specific formulas (Unmanned Aerial Vehicles), while we focus on a generic, domain-independent solutions, such as Mars Rovers, Blockworlds, Health, Fire-extinction, etc. Furthermore, their approach specifications are given manually, while we aim

at offering a great degree of automation by providing templates. While they build on a domain description language based on TAL (Temporal Action Logic), we build on the standardized language PDDL.

In this paper, we present a first approach to explore the benefits of using techniques from the runtime verification community to alleviate the problems that arise in the field of plan monitoring at its execution time. While these two fields are typically seen as disconnected from each other, runtime verification provides a formal methods based backend to automatically generate executable monitors for a given plan. We suggest that this link is worth exploring and therefore advocate the idea of deploying runtime verification as a technique for plan execution monitoring. We sidestep the cumbersome task of compiling a specification for the monitoring goals by providing a set of generic templates, which allow to express temporal properties involving timing requirements. These templates are instantiated and configured through a set of parameters we automatically derive from the plan. Finally, we use results from the runtime verification community to automatically derive an executable observer. We show how this technique seamlessly integrates into the traditional planning approach, by implementing the technique into a multi-agent domain-independent planning, execution and monitoring architecture. In this paper we do not explain the main components of this architecture we only focus on the monitoring component.

2 Motivating Example: Mars Rovers

In this section, we elaborate on a possible Mars Rovers scenario¹ to detail the problem of monitoring and executing autonomous plans for a dynamic multi-agent system. In this system, two agents (i.e., rovers) are working in close proximity and need to cooperate in order to accomplish a mission.

The scenario assumes that there are two Mars Rovers (R_a and R_b) and three waypoints $\{W_1, W_2, W_3\}$, where W_2 is the waypoint of the Lander L and the initial state of both rovers. Rovers R_a and R_b are working near each other, both equipped with a set of navigation cameras. R_a uses a microscopic camera to analyze rocks in waypoint W_1 and sends the results to the lander, whereas R_b takes panoramic pictures of the surrounding terrain and communicates it to the lander in the waypoint W_2 . Communication with the lander and the satellite can be initiated from any waypoint W_x . The rovers are constrained by limited energy, on-board data storage, downlink opportunities, as well as available bandwidth and time to complete observations.

These limitations complemented with the unpredictability of the environment, may cause unforeseen problems during the mission. In the following, we will show how our template based plan execution monitoring helps to detect such failures as early as possible.

The initial plan for the mission is provided from a planner agent, i.e., the control center on Earth. In a subsequent step, each plan is forwarded to a dedicated

¹ We work with the rover temporal domain as defined in the International Planning Competition (IPC).

decision support module, located in the planner agent, to derive the variables to be monitored during plan execution. The goal is to determine the information that needs to be monitored to guarantee a successful plan execution. We use an anytime regression approach to *automatically* select the variables to be observed by the monitoring module during the plan monitoring. We compute the variables to be monitored through an extension of the goal regression method proposed in [11]. In the following, we refer to this information as *monitor-parameters*.

Then, this plan is divided into a set of actions, which are, together with their respective goals, and their *monitor-parameters* sent to both rovers R_a and R_b to be executed. Each of the rovers executes its assigned plan which consists of sequentially executed actions. Let's suppose the first action planned to be executed by R_a is the action *Navigate*. The action navigates the rover from waypoint W_2 to waypoint W_1 within a scheduled duration of 10 time units.

The following are the *high-level* conditions and effects for the action *Navigate*, as dictated by the domain:

$$\begin{aligned} \text{preconditions} &:= \left\{ \begin{array}{l} \text{at}(\text{Ra}, W_2) \wedge \\ \text{energy}(\text{Ra}) \geq 8 \wedge \\ \text{available}(\text{Ra}). \end{array} \right. \\ \text{invariants} &:= \left\{ \begin{array}{l} \text{can_traverse}(\text{Ra}, W_2, W_1) \wedge \\ \text{visible}(W_2, W_1). \end{array} \right. \\ \text{effects} &:= \left\{ \begin{array}{l} \text{at}(\text{Ra}, W_1) \wedge \\ \text{energy}(\text{Ra}) - = 8. \end{array} \right. \end{aligned}$$

The three preconditions need to hold *before* executing the action *Navigate* and the two invariants need to hold *throughout* the execution of *Navigate*. The effect describes the change in the rover's state *after* executing the action *Navigate*.

In real plan execution, the aim of the monitoring module is to verify that the action is executable. A dedicated monitoring module thus checks whether the preconditions and the invariants hold in the current state. All this information is kept in the *monitor-parameters*. Formally, we define the *monitor-parameters* as a tuple $\langle \mathcal{L}, \mathcal{T}, \mathcal{V} \rangle$, where:

- \mathcal{L} is a tuple $\langle L_v, L_i \rangle$ with the set of variables to be monitored (L_v) and an identifier (L_i) to express whether the variable is an invariant I , a precondition C , or an effect E , i.e., those that are directly related to the plan.
- \mathcal{T} is a tuple with the time at which the variable is generated (t_g), and the earliest (t_e) and latest (t_l) time at which the variable will be used.
- \mathcal{V} is the value range for each variable, denoting the set of correct values that the variables can take on.

Table 1, shows the *monitor-parameters* for the action *Navigate* of R_a . For example, the last entry in the table describes the effect of executing *Navigate* on the energy level of the rover, i.e., the energy level will decrease by 8 units after execution.

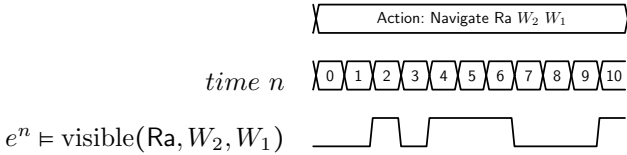
Table 1. Monitor-parameters of action *Navigate*

L_v	L_i	t_g	t_e	t_l	V
<code>can_traverse(Ra, W2, W1)</code>	I	0	0,01	10,01	true
<code>visible(W2, W1)</code>	I	0	0,1	10,1	true
<code>at(Ra, W2)</code>	C	0	0,1	0,1	true
<code>available(Ra)</code>	C	0	0,01	10,1	true
<code>energy(Ra)</code>	C	0	0,01	0,1	$[8, \infty)$
<code>at(Ra, W1)</code>	E	10,1	10,1	20,1	true
<code>energy(Ra) decreasing</code>	E	10,1	10,1	10,1	8

3 From Plan Execution Monitoring to Runtime Verification

In this section, we briefly summarize the foundations of runtime verification and real-time temporal logics which we will use to express templates in our framework. For further details, we refer the reader to more elaborates sources such as [2,5].

Real-time systems (such as the Mars Rovers) often do not only need to comply with a set of functional requirements but also – equally important – with tight timing constraints. Thus, the underlying logics need to allow to reason about certain timing assumptions. We will illustrate the concept of runtime verification along the following example:



which shows the validity of $e^n \models \text{visible}(\text{Ra}, W_2, W_1)$, for the prefix of execution e , where $\text{visible}(\text{Ra}, W_2, W_1)$ is a proposition over the state of the system. This information is incrementally collected during a run of the system, i.e., while executing the action *Navigate*.

Executions. Let $e = (s_t)_{t \geq 0}$ be the (sequential) execution of a set of actions $a \in \mathcal{A}$ where s_t is a state of the system at time t . A proposition $p \in \mathcal{P}$ holds on s_t iff $p \in s_t$. Denote by e^n , for $n \in \mathbb{N}_0$, the *execution prefix* $(s_t)_{0 \leq t \leq n}$. For example *Navigate*, *Takelimage* $\in \mathcal{A}$ and $\{\text{energy}(\text{Ra}) \geq 8, \text{visible}(\text{Ra}, W_2, W_1), \dots\} \in \mathcal{P}$. In the running example, W_1 is visible for R_a at times $n \in \{2, 4, 5, 6, 10\}$ (we write $e^n \models \text{visible}(\text{Ra}, W_2, W_1)$), whereas W_1 is not visible at times $n' \in \{1, 3, 7, 8, 9\}$ we write $e^{n'} \not\models \text{visible}(\text{Ra}, W_2, W_1)$.

Temporal Logics. In runtime verification a formal specification formalism is used to specify correctness claims over an execution. A popular formalism is linear temporal logic (LTL) and it's real-time extension metric temporal logic (MTL).

MTL [1] extends LTL by replacing the qualitative temporal modalities of LTL by quantitative modalities that respect time bounds. The syntax of MTL is defined as follows. For every atomic proposition $p \in \mathcal{P}$, p is a formula. $J = [t, t']$ describes a time interval for some $t, t' \in \mathbb{N}_0$. If φ and ψ are formulas, then so are:

$$\neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \varphi U_J \psi$$

For a MTL formula ξ , time $n \in \mathbb{N}_0$ and execution e , we define ξ holds at time n of execution e , denoted $e^n \models \xi$, inductively as follows:

$$\begin{aligned} e^n \models \text{true} & \quad \text{is true,} \\ e^n \models \Sigma & \quad \text{iff } \Sigma \in AP \text{ holds in } s_n, \\ e^n \models \neg\varphi & \quad \text{iff } e^n \not\models \varphi, \\ e^n \models \varphi \bullet \psi & \quad \text{iff } e^n \models \varphi \bullet e^n \models \psi \text{ with } \bullet \in \{\wedge, \vee, \rightarrow\}, \\ e^n \models \varphi U_J \psi & \quad \text{iff } \exists i (i \geq n) : (i - n \in J \wedge e^i \models \psi \wedge \forall j (n \leq j < i) : e^j \models \varphi). \end{aligned}$$

The Until within interval modality $\varphi U_J \psi$ allows to encode timed relationships between two formulas, with the intended meaning: φ needs to hold continuously until (at some time within J) ψ holds. The time bounds described by J are relative to the current time n . AP denotes the set of atomic propositions of the formula. In our case we use conjunctions of inequalities over variables to be monitored. Naturally, a formula ξ satisfies execution e , denoted $e \models \xi$, iff for all $i \in \mathbb{N}_0$, it holds that $e^i \models \xi$. With the dualities [4]

$$\begin{aligned} \text{true } U_J \phi & \equiv \exists \vec{J} \varphi \\ \neg \exists \vec{J} \neg \phi & \equiv \forall \vec{J} \varphi \end{aligned}$$

we arrive at two additional modalities: $\exists \vec{J} \varphi$ with the intended meaning φ needs to hold eventually within the interval J as well as $\forall \vec{J} \varphi$ interpreted as φ is an invariant within interval J . Technically, a single observer capable of monitoring $\varphi U_J \psi$ is sufficient to evaluate any of $\forall \vec{J} \varphi, \exists \vec{J} \varphi$ claims, as we can always rewrite them by the equivalences above to $\varphi U_J \psi$. We argue that this fine grained breakdown of the until modality has the following advantages: (a) Allows to directly map frequently occurring claims to a modality and (b) yields more efficient observers compared to instantiating a full-fledged $\varphi U_J \psi$ observer (anonymous).

The Runtime Verification Problem. Having defined executions and temporal logics, we can define a runtime verification problem as: *Given a temporal logic formula ξ and an execution e , does $e^n \models \xi$ for $n \in \mathbb{N}_0$ hold?*

Monitors. Checking whether a MTL formula holds at time $n \in \mathbb{N}_0$ in some execution $e = (s_t)_{t \geq 0}$ can be determined by results from the current state s_n and its successor states s_{n+1} . For example, evaluating the invariant $\xi = \forall_{[4,6]} \text{visible}(\text{Ra}, W_2, W_1)$ on execution $e = (s_t)_{t \geq 0}$ requires to check:

$$e^n \models \xi \Leftrightarrow \bigwedge_{t=4}^6 (\text{visible}(\text{Ra}, W_2, W_1) \text{ holds on } s_t)$$

Note that the problem of monitoring $e^n \models \xi$ has been studied extensively in the past; see [18] for a survey. Thus, efficient algorithms to decide $e^n \models \xi$ are not an aim of this paper. For our implementation, we make use of existing algorithms. Possible choices include: a) translate the temporal formula into a finite-state automaton that accepts all the models of the specification. The translation may be based on an on-the-fly adaption of the tableau construction [13,21] or make use of timed automata [8,16], b) restricting MTL to its *safety* fragment and to wait until the bounded future operators have elapsed and decide validity afterwards [17,7] and c) restricting temporal logics to its *past-time* fragment [19,6,16,8]. The conditions and effects for the action `Navigate` can be captured in MTL:

$$\begin{aligned}\varphi_1 &:= (n == 0) \rightarrow \forall_{[0,10]}^{\rightarrow} (\text{can_traverse}(\text{Ra}, W_2, W_1) \wedge \text{visible}(\text{Ra}, W_2, W_1)) \\ \varphi_2 &:= (n == 0) \rightarrow (\text{at}(\text{Ra}, W_2, W_1) \wedge \text{energy}(\text{Ra}) \geq 8 \wedge \text{available}(\text{Ra}, W_2, W_1)) \\ \varphi_3 &:= (n == 10) \rightarrow (\text{energy}(\text{Ra}) - = 8)\end{aligned}$$

Property φ_1 captures the invariant condition for the rover execution of action `Navigate`. Informally, it requires that for all times $n' \in [0,10]$ the predicates `can_traverse(Ra, W2, W1)` and `visible(Ra, W2, W1)` need to hold. Property φ_2 checks for the preconditions: At time $n = 0$, `(at(Ra, W2, W1) \wedge energy(Ra) \geq 8 \wedge available(Ra, W2, W1))` needs to hold. Property φ_3 checks for the effect of the action: At time $n = 10$, make sure that the energy decreases by 8 units. The force of capturing conditions and effects of an action is that the problem of plan execution monitoring can now be translated into a runtime verification problem: *Decide whether $\varphi_1 \wedge \varphi_2 \wedge \varphi_3$ holds on execution e , i.e., does $e \models \varphi_1 \wedge \varphi_2 \wedge \varphi_3$?*

4 Template-Based Synthesis of Plan Execution Monitors

In this section, we will provide a generalization of the MTL encodings of the conditions and effects associated with the action `Navigate` in the Mars Rover example. We give an algorithm that allows to parametrize the templates according to the monitor-parameters derived from the *decision support module*.

We start with four templates that allow to encode arbitrary monitor-parameter $\langle \mathcal{L}, \mathcal{T}, \mathcal{V} \rangle$ into an MTL specification. The first one encodes invariants and their associated time bounds whereas the second one encodes preconditions. The third one encodes the effects. And the final template ensures that for a single action, the conjunction of all invariants and preconditions needs to hold.

Template 1: Invariants

$$\varphi_I := (n == T_a) \rightarrow \forall_{[T_b, T_c]}^{\rightarrow} \left(\bigwedge_{i \in T_a} \text{pred}(i) \right)$$

Template 2: Preconditions

$$\varphi_P := (n == T_a) \rightarrow \left(\bigwedge_{i \in T_b} \text{pred}(i) \right)$$

Template 3: Effects

$$\varphi_E := ((n == T_a) \rightarrow (\bigwedge_{i \in T_b} \text{pred}(i))) \wedge ((n == T_c) \rightarrow (\bigwedge_{j \in T_d} \text{pred}(j)))$$

Template 4: Consistency

$$\varphi_C := \bigwedge_{i \in I} \varphi_I^i \wedge \bigwedge_{j \in P} \varphi_P^j \wedge \bigwedge_{k \in E} \varphi_E^k$$

The templates from above are instantiated by Algorithm 1. The number of formulas generated is linear in the number of $\langle \mathcal{L}, \mathcal{T}, \mathcal{V} \rangle$ tuples in the monitor-parameters, i.e., six in the case of the action `Navigate`.

Algorithm 1. Template-based Synthesis

```

1: Let  $\varphi_C$  be the resulting MTL encoding
2: for each  $\langle \mathcal{L}, \mathcal{T}, \mathcal{V} \rangle$  in monitor parameters do
3:   if  $\mathcal{L}.L_i$  is of type Invariant then
4:      $\varphi_I^i \leftarrow (n == \mathcal{T}.t_e) \rightarrow \forall_{[0, (\mathcal{T}.t_i - \mathcal{T}.t_e)]}(\mathcal{L}.L_v)$ 
5:   end if
6:   if  $\mathcal{L}.L_i$  is of type Precondition then
7:      $\varphi_P^j \leftarrow (n == \mathcal{T}.t_e) \rightarrow (\mathcal{L}.L_v)$ 
8:   end if
9:   if  $\mathcal{L}.L_i$  is of type Effect then
10:     $\varphi_E^k \leftarrow (n == \mathcal{T}.t_e) \rightarrow (\mathcal{L}.L_v) \wedge (n == \mathcal{T}.t_i) \rightarrow (\mathcal{L}.L'_v)$ 
11:   end if
12: end for
13:  $\varphi_C \leftarrow \bigwedge_{i \in I} \varphi_I^i \wedge \bigwedge_{j \in P} \varphi_P^j \wedge \bigwedge_{k \in E} \varphi_E^k$ 
14: return  $\varphi_C$ 

```

Fig. 1 shows the integration of our template-based synthesis approach into the multi-agent framework. Inputs to the decision support module (DSM) are the generated Plan and the Domain. The DSM then derives, through an anytime regression approach, a set of monitor-parameters $\langle \mathcal{L}, \mathcal{T}, \mathcal{V} \rangle$ which are the input to the template-based synthesis algorithm. Based on templates above, this step yields the MTL encoding φ_C of the preconditions and invariants which are required to hold during execution of the plan. In the final step, we instantiate the algorithms described by [19], to derive a monitor for $e^n \models \varphi_C$. Please note that, the process of generating the monitor is fully automatic, with no manual intervention or modelling required.

The resulting monitor is fully executable and might run as a dedicated software task at the agent's software stack, or, alternatively be compiled into executable hardware blocks, i.e., targeting a field-programmable gate array (FPGA) or an application-specific integrated circuit (ASIC) platform. ASIC has the advantage that monitoring can be performed external to the agent, i.e., without interfering with it's runtime behavior.

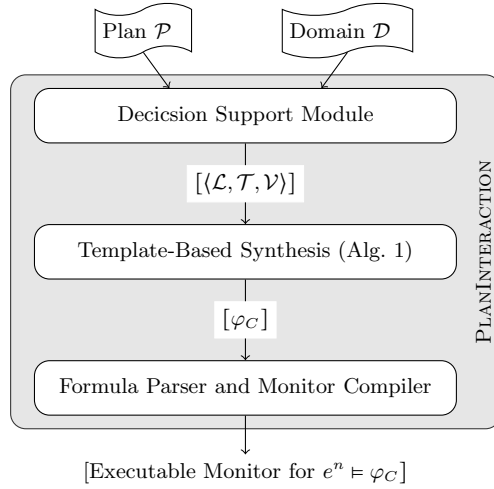


Fig. 1. Toolchain integration into the PLANINTERACTION framework

5 Templates for Continuous Monitoring

We now discuss further templates which allow to encode deep relationships among physical processes of variables. We discuss several optimizations to keep the resulting monitors small.

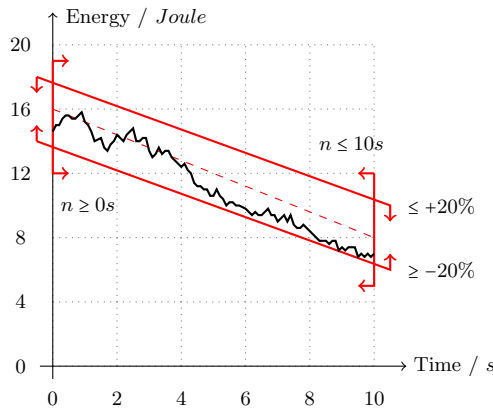


Fig. 2. Continuous monitoring the effect energy(Ra)– = 8 of action Navigate

Consider the scenario depicted in Fig. 2, where the energy plot describes a discharge characteristic typical for the action Navigate. The exact discharge plot is different from run to run as it is influenced by factors hidden to the high level planner. Unforeseen changes in the terrain or sensor noise require to allow for some kind of safety margin describing the allowed range for the battery

drain during movement. Therefore, a monitoring strategy which continuously evaluates $\text{energy}(\text{Ra})$ and compares the result against $E(n) = -0.8 \cdot n + 16$ is infeasible in a real-life setting. Ideally, we want to monitor a weaker version of $E(n) = -0.8 \cdot n + 16$, such as the two-dimensional convex polyhedron bounded by the conjunction of the following inequalities:

$$\varphi_l := \begin{cases} n \geq 0 \wedge \\ n \leq 10 \wedge \\ E(n) \geq 80\% \cdot (-0.8 \cdot n + 16) \wedge \\ E(n) \leq 120\% \cdot (-0.8 \cdot n + 16). \end{cases}$$

We can generalize from these constraints and derive another template that allows to encode linear decreasing or increasing relationships, such as the constraints over the energy level of the rover (Template 5). Again T_a, T_b, T_c are directly created from the monitor-parameters $\langle \mathcal{L}, \mathcal{T}, \mathcal{V} \rangle$ and $\text{predicates}(T_d)$ selects the predicate returning the variable to be monitored. $T_f \in [0, 1]$ allows to specify the safety margin, whereas T_e is the slope of the expected increasing or decreasing characteristics and T_g is the result of evaluating $\text{predicates}(T_d)$ at time $n = T_a$.

Template 5: Linear decreasing/increasing with Safety Margin

$$\varphi_L := (n == T_a) \rightarrow \forall_{[T_b, T_c]}^{\rightarrow} (\text{pred}(T_d) \leq (1 + T_f) \cdot (T_e \cdot n + T_g) \wedge \text{pred}(T_d) \geq (1 - T_f) \cdot (T_e \cdot n + T_g))$$

6 Conclusion and Future Work

We have studied the problem of automatically synthesizing run-time observers for plan execution monitoring from the domain description and the scheduled plan. Monitor generation works in two phases: First, we use a set of templates to automatically compile an equivalent encoding in metric temporal logic (MTL) of the constraints to be monitored. Second, we use algorithms known from the runtime verification community to compile efficient runtime monitors. The monitors are efficient in the sense that monitoring allows to include timing constraints and detects deviations from the scheduled plan as early as possible. This yields greater flexibility for a high-level re-planner module.

As future work, we plan to conduct an industrial case-study to demonstrate the benefits of our approach in real-life planning scenarios. Additionally, we plan to evaluate the use of probabilistic reasoning (for example Bayesian Networks, as in [20]) to assess the likelihood of some pre-defined error hypothesis. This would not only help us to detect deviations from the plan at mission time, but also provide reasoning about the root cause of the detected malfunction. In this setting, intermediate outputs of our runtime monitors serve as input to the reasoning module. We believe that this information would be valuable for a high-level re-planner module.

References

1. Alur, R., Henzinger, T.A.: Real-time Logics: Complexity and Expressiveness. In: LICS, pp. 390–401. IEEE (1990)
2. Alur, R., Henzinger, T.: Logics and models of real time: A survey. In: Huizing, C., de Bakker, J.W., Rozenberg, G., de Roever, W.-P. (eds.) REX 1991. LNCS, vol. 600, pp. 74–106. Springer, Heidelberg (1992)
3. Ambros-Ingerson, J.A., Steel, S.: Integrating planning, execution and monitoring. In: Proceedings of the AAAI, pp. 83–88 (1988)
4. Baier, C., Katoen, J.P.: Principles of Model Checking. The MIT Press (2008)
5. Barringer, H., Falcone, Y., Finkbeiner, B., Havelund, K., Lee, I., Pace, G., Roşu, G., Sokolsky, O., Tillmann, N. (eds.): RV 2010. LNCS, vol. 6418. Springer, Heidelberg (2010)
6. Basin, D., Klaedtke, F., Zălinescu, E.: Algorithms for monitoring real-time properties. In: Khurshid, S., Sen, K. (eds.) RV 2011. LNCS, vol. 7186, pp. 260–275. Springer, Heidelberg (2012)
7. Basin, D.A., Klaedtke, F., Müller, S., Pfitzmann, B.: Runtime monitoring of metric first-order temporal properties. In: FSTTCS, pp. 49–60 (2008)
8. Divakaran, S., D’Souza, D., Mohan, M.R.: Conflict-tolerant real-time specifications in metric temporal logic. In: TIME, pp. 35–42 (2010)
9. Erann Gat, J.F., Miller, D.: Planning for execution monitoring on a planetary rover. In: Proceedings of the Space Operations Automation and Robotics Workshop (1990)
10. Fikes, R.E., Hart, P.E., Nilsson, N.J.: Learning and executing generalized robot plans. In: Readings in Knowledge Acquisition and Learning, pp. 485–503. Morgan Kaufmann Publishers Inc. (1993)
11. Fritz, C., McIlraith, S.A.: Monitoring plan optimality during execution. In: ICAPS, pp. 144–151 (2007)
12. Gat, E., Slack, M.G., Miller, D.P., Fiby, R.: Path planning and execution monitoring for a planetary rover. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 20–25 (1990)
13. Geilen, M.: An improved on-the-fly tableau construction for a real-time temporal logic. In: Hunt Jr., W.A., Somenzi, F. (eds.) CAV 2003. LNCS, vol. 2725, pp. 394–406. Springer, Heidelberg (2003)
14. Gianni, M., Papadakis, P., Pirri, F., Liu, M., Pomerleau, F., Colas, F., Zimmermann, K., Svoboda, T., Petricek, T., Kruijff, G.J., Khambhaita, H., Zender, H.: A unified framework for planning and execution-monitoring of mobile robots. In: PAMR. AAAI, AAAI Press (August 2011)
15. Kvarnström, J., Heintz, F., Doherty, P.: A temporal logic-based planning and execution monitoring system. In: ICAPS, pp. 198–205 (2008)
16. Maler, O., Nickovic, D., Pnueli, A.: Real time temporal logic: Past, present, future. In: Pettersson, P., Yi, W. (eds.) FORMATS 2005. LNCS, vol. 3829, pp. 2–16. Springer, Heidelberg (2005)
17. Maler, O., Nickovic, D., Pnueli, A.: On synthesizing controllers from bounded-response properties. In: Damm, W., Hermanns, H. (eds.) CAV 2007. LNCS, vol. 4590, pp. 95–107. Springer, Heidelberg (2007)

18. Maler, O., Nickovic, D., Pnueli, A.: Checking temporal properties of discrete, timed and continuous behaviors. In: Avron, A., Dershowitz, N., Rabinovich, A. (eds.) *Trakhtenbrot/Festschrift*. LNCS, vol. 4800, pp. 475–505. Springer, Heidelberg (2008)
19. Reinbacher, T., Függer, M., Brauer, J.: Real-time runtime verification on chip. In: Qadeer, S., Tasiran, S. (eds.) *RV 2012*. LNCS, vol. 7687, pp. 110–125. Springer, Heidelberg (2013)
20. Schumann, J., Mengshoel, O.J., Srivastava, A.N., Darwiche, A.: Towards software health management with Bayesian networks. In: *FoSER*, pp. 331–336 (2010)
21. Thati, P., Roşu, G.: Monitoring Algorithms for Metric Temporal Logic specifications. *ENTCS* 113, 145–162 (2005)

Distributed Privacy-Preserving Minimal Distance Classification

Bartosz Krawczyk and Michał Woźniak

Department of Systems and Computer Networks,
Wrocław University of Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{bartosz.krawczyk,michal.wozniak}@pwr.wroc.pl

Abstract. The paper focuses on the problem of preserving privacy for a minimal distance classifier working in the distributed environment. On the basis of the study of available works devoted to privacy aspects of machine learning methods, we propose the novel definition and taxonomy of privacy. This taxonomy was used to develop new effective classification algorithms which can work in distributed computational environment and assure a chosen privacy level. Instead of using additional algorithms for secure computing, the privacy assurance is embedded in the classification process itself. This lead to a significant reduction of the overall computational complexity what was confirmed by the computer experiments which were carried out on diverse benchmark datasets.

Keywords: privacy preserving, distributed data mining, classification, k -NN.

1 Introduction

Nowadays most of the enterprizes have collected huge amounts of valuable data, unfortunately their manual analysis is virtually impossible. The market-leading companies realize that smart analytic tools which are capable to interpret collected data could lead to business success. Therefore they desire to exploit strengths of machine learning techniques to extract hidden, valuable knowledge from the huge, usually distributed databases. Classification methods are applied in many practical areas [7] as banking, security, marketing to name only a few. Numerous approaches have been proposed to construct efficient pattern recognition methods [1], but in this work we focus on the k -nearest neighbor rule [6], which is one of the most fundamental and simplest nonparametric classification algorithms. The minimal distance classification is attractive from the theoretical and the practical point of view, because it is the recommended approach for discriminant analysis when the knowledge about probability densities is insufficient. However, its theoretical properties guaranteeing that for all distributions the probability error is bounded above by twice the Bayes probability error [5]. Additionally, the advantage of the naive implementation of this rule is that it

does not have a learning phase, which protects it against unwanted phenomena, such as overfitting [1].

Let us revert to the subject of the privacy preserving distributed data analysis. We should notice using distributed databases could come up against legal or commercial limitations which do not allow sharing raw data between databases or merging them in the common repository. Therefore, developing privacy preserving versions of the lived-in data analysis techniques is the focus of intense research. The aim of privacy preserving is to offer a possibility of distributed data analysis with a minimum of information disclosure.

2 Privacy Preserving

Westin [12] circumscribes the privacy as "control over personal information". Unfortunately such a definition is too general to be used in practice, therefore Moor [10] notices that the concept of privacy itself is best explained in terms of restricted access, not control. It is worth noting another related problem of individual privacy in public spaces, which is also widely discussed [11]. Privacy preserving data mining methods focus on hidden information related to an individual record. One of the most popular approaches uses data perturbation techniques or secure models to protect privacy of distributed repositories [8]. Some general tools supporting privacy-preserving data analysis as the secure sum, comparison, division, or scalar product are proposed and evaluated in many articles [4].

Most of the works on privacy preserving data mining show the aspect of privacy as a binary one - full privacy or lack of privacy. In our opinion we can distinguish several stages of privacy, what encourages us to formulate a new privacy definition. Each peer has data and view.

Definition 1. *Data means raw data stored in a given database.*

Definition 2. *View means an information about data stored in a given database e.g., statistic or the lowest distance between a given object and an object stored in the database.*

Definition 3. *Privacy preserving means that the view of the other side is restricted to the data marked as public.*

Definition 4. *Lack of privacy means that the view of the other node allows access to private part of the database, thus creating threat to its owner.*

We propose a novel taxonomy for splitting the privacy into several levels. With each of the levels there is a corresponding table showing what range of privacy is assured. Plus denotes public data, while minus stands for private information.

1. **No privacy (NP)** means that all data (an views) are public i.e., we can identify which data is possessed by which database.
2. **Local data privacy (LDP)** means that all data are public but we are not able to identify which data is possessed by which database i.e., we know that a given object (record) is stored in one of the databases.
3. **Global data privacy (GDP)** means that data are private but we can get views about a given databases.
4. **Local view privacy (LVP)** means that views are public but we are not able to identify which views of which database is presented i.e., we know that a given view is a view of one of the databases.
5. **Full privacy (FP)** means that neither data nor views are available.

From the practical point of view the GDP and LVP levels are interesting. For the NP and LDP level we can merge datasets and use a classical method to mine the accumulative database. For the FP sharing any information is not allowed, therefore we can not use any information derived from individual dataset i.e., we can mine database in our disposal only.

Our proposal concentrates on preserving the privacy of a source database / node. By this we conceal what data is in possession of a side participating in a collaborative classification. This is an important aspect in many real-life domains e.g., medical informatics or social network analysis.

The comparisons of different type of privacy are presented in Tab.1.

Table 1. Comparisons of different privacy levels

	data		view	
	local	global	local	global
<i>NP</i>	+	+	+	+
<i>LDP</i>	-	+	-	+
<i>GDP</i>	-	-	+	+
<i>LVP</i>	-	-	-	+
<i>FP</i>	-	-	-	-

3 Proposed Privacy Preserving Modifications of k -NN Classifier

In our research we focus on the possibilities of protecting data by mechanisms that are offered by classification algorithms themselves. Usually, privacy preserving methods are based on supplementary mechanisms like Yao's protocol [9]. Unfortunately, they usually require extra remarkable computational cost, therefore solutions that use mechanisms embedded in the classification process itself are more interesting.

The k -nearest neighbour classifier offers interesting possibilities for task of privacy protecting data mining. It is worth noting that even in its basic form this algorithm primitively conceals data itself, because the object is not used in

the classification step, but only the value of its distance to classified pattern. Therefore, we can find the hypersphere on which this object is located, but we cannot guess its value directly. We assume that in the described process we have \mathbf{V} different databases (partitions, sides, nodes) connected in a distributed environment e.g., cloud.

On the basis of the taxonomy presented in the previous section, four different approaches for privacy preserving k -NN in the distributed environment are proposed. Each of them has its own advantages and drawbacks. It is up to the individual user to choose the scheme most adjusted to his/her needs.

Algorithm 1. Normal querying with full neighbour set

- 1: $\mathbf{Vs} \rightarrow$ set of nodes, $\mathbf{V} \rightarrow$ number of nodes,
 - 2: $\mathbf{k} \rightarrow$ number of neighbors (parameter) of k -NN algorithm
 - 3: query all nodes from \mathbf{Vs} for \mathbf{k} -nearest neighbours
 - 4: main node sorts received objects
 - 5: the best (nearest) \mathbf{k} objects are chosen
-

Benefits : Each node sends the same number of objects, therefore none of them has the knowledge which of them are used (**local data privacy level achieved**).

Drawbacks : Each node sends set of k objects. Therefore in more complex problems it reveals some direct information.

Algorithm 2. Ranked querying

- 1: $\mathbf{Vs} \rightarrow$ set of nodes, $\mathbf{V} \rightarrow$ number of nodes
 - 2: $\mathbf{k} \rightarrow$ number of neighbors (parameter) of k -NN algorithm
 - 3: $\mathbf{t} = 0$
 - 4: $\mathbf{n} \rightarrow$ specified by user, $\mathbf{n} \leq \mathbf{V}$
 - 5: **while** $\mathbf{t} == \mathbf{k}$ or no better solution found **do**
 - 6: query all nodes from \mathbf{Vs} for nearest neighbor
 - 7: remove \mathbf{n} nodes with the worst responses from \mathbf{Vs} set
 - 8: $\mathbf{t} = \mathbf{t} + 1$
 - 9: **end while**
-

Benefits : Number of queries is reduced. Databases reveal significantly less information, because some of them are eliminated from querying during the procedure (**global data privacy level achieved**).

Drawbacks : Database that was eliminated in the beginning is informed that its objects are not participating in the classification. Database that was asked frequently may assume that its objects play a major role in decision process.

Algorithm 3. Set of best objects

```

1:  $\mathbf{Vs}$   $\rightarrow$  set of nodes,  $\mathbf{V}$   $\rightarrow$  number of nodes
2:  $\mathbf{k}$   $\rightarrow$  number of neighbors (parameter) of k-NN algorithm
3: generate set of  $\mathbf{k}$  random objects in the main node
4: denote them as the set of best objects
5: while  $V \neq 0$  do
6:   choose random node  $\mathbf{Vr}$  from  $\mathbf{Vs}$  set and send the set of best objects to the
     chosen node
7:   compare objects in the node with the received set
8:   replace all worse objects
9:   remove this node from  $\mathbf{Vs}$  set
10:  return the set to the main node
11:  denote it as the set of best objects
12:  remove the node from the set  $V [V = V - Vr]$ 
13: end while

```

Benefits : Due to the random set generation in the first step none of the nodes knows if it is asked as the first or the last one. Therefore, a database can not guess the information that the package of objects it sends to the main node is the final set in the classification process. Also nodes are unable to identify to which databases the objects from the input set belongs. Each node reveals only the objects that are sent to the main node. If the database does not have better objects, none of them are revealed and it is impossible to point out which database does not return any new objects. (**local view privacy level achieved**).

Drawbacks : This approach assumes that each node compares on its side the whole received set with objects stored in the database. Therefore the computational complexity of this algorithm is higher than the previous ones.

Benefits : Due to random set generation in the first step none of the nodes knows if it is asked as the first or the last one. Therefore any database can not guess whether the information it sends to the main node is the final set in the classification process. Also, they are unable to identify to which databases the objects from the input set belong to. Each node reveals only the objects that are sent to the main node. If the database does not have better objects, none of them are revealed and it is impossible to indicate which database does not return new objects. Additionally the number of queries is significantly reduced – if tested database can not return better objects then the query ends. Each node does not receive the tested set but only one object from it at the time. If there are no better objects in the database it will not receive more objects, thus concealing their values (**local view privacy level achieved**).

Drawbacks : This approach assumes that each node has to compare on its side received objects with objects stored in the database. Therefore, the computational complexity is the highest, compared to the previously presented methods. Additionally, if a query from the main node ends quickly, this side may assume that its objects will not participate in the classification process.

Algorithm 4. Step querying with increasing neighbour set

```

1:  $\mathbf{Vs}$   $\rightarrow$  set of nodes,  $\mathbf{V}$   $\rightarrow$  number of nodes
2:  $\mathbf{k}$   $\rightarrow$  number of neighbors (parameter) of  $k$ -NN algorithm
3:  $\mathbf{z}$   $\rightarrow$  query parameter
4: generate set of  $\mathbf{k}$  random objects in the main node
5: denote them as the set of best objects
6: while  $\mathbf{V} == 0$  do
7:   choose random node from  $\mathbf{Vs}$  set
8:    $\mathbf{z} = \mathbf{k}$ 
9:   while no better objects found or  $\mathbf{z} = 0$  do
10:    send object number  $\mathbf{z}$  from the best set to this node and compare it with
    objects in the node
11:    if there is a better one send it to the main node then
12:       $\mathbf{z} = \mathbf{z} - 1$ 
13:    end if
14:  end while
15:  the main node sorts the set of received objects and denote them as the set of
  best objects
16:  remove this node from  $\mathbf{Vs}$  set
17:   $\mathbf{V} = \mathbf{V} - 1$ 
18: end while
19: if final set of best objects  $==$  set of random objects then
20:   repeat the procedure
21: end if

```

4 Experimental Evaluation

Usually the three performance metrics for privacy preserving classification algorithm are used [3]:

1. Accuracy - which evaluates the loss of accuracy of privacy preserving method compared with the original one.
2. Efficiency - which evaluates the computational and memory complexity of privacy preserving modification compared with the original algorithm.
3. Privacy - which estimates how much information is shared between the nodes during privacy preserving modification.

As we mentioned in the previous section in this work we focus on the privacy mechanisms that are offered by original k -NN method. Therefore, our propositions are as accurate as the original k -NN and each of them assures the appropriate privacy level what is discussed in the previous section. The main goal of the experiments is to evaluate the computational complexity of the proposed methods. To provide the most accurate and exhaustive tests of the proposed methods we had carefully selected the set of diverse benchmark databases. During the tests we would like to evaluate dependencies between the number of distributed database partitions and the number of required neighbors k , which is the crucial

parameter of the algorithm under consideration. Therefore, we decided to carry out the tests for different values of k and the number of partitions V . We had assumed that each dataset has a permanent number of objects and they had been divided equally between V nodes.

4.1 Set-Up

To examine the behavior of the proposed methods on most diverse benchmarks we chose a high dimensionality, small sample size set, two very numerous sets with small number of features and a typical one, what allowed us to cover the wide range of real-life possibilities and make our tests more practically-oriented. All datasets come from UCI Machine Learning Repository [2] and they are described in the Tab.2.

All experiments were carried out on Intel Core Duo T5800 2,0 GHz CPU with

Table 2. Detailed description of four data sets used for evaluation of the proposed methods.

<i>Dataset name</i>	<i>Arcene</i>	<i>Adult</i>	<i>Letter</i>	<i>Splice – junction</i>
<i>Number of objects</i>	900	48842	20000	3190
<i>Number of features</i>	10000	14	16	61

3 GB RAM memory in R environment, with k -NN algorithm taken from dedicated package, thus ensuring that results achieved the best possible efficiency and that performance was not decreased by bad implementation. For comparison we chose the following methods described in the previous section: (**A4-1**) normal querying with full neighbor set, (**A4-2**) Ranked querying, (**A4-3**) Set of the best objects, (**A4-4**) Step query with increasing neighbor set.

4.2 Result

Figures from 1 to 4 show the execution time of proposed k -NN modifications for each of the databases.

Some interesting observations can be made on the basis of the experimental results.

- The overall maximum classification time did not exceed 15 minutes for very numerous dataset (circa 50000 objects) when using the most secure method introduced in this paper. Other approaches very rarely include their computational times, but in all presented cases they are at least a few times greater.
- The computational time decrease according to the increase of the number of nodes. This reduction is significant for the small number of nodes (2–4). Further increasing number of nodes does not result in significant gain of the computational time.

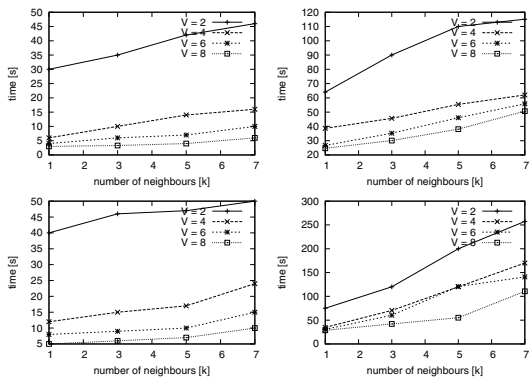


Fig. 1. Time complexity for the Arcene dataset (clockwise, starting from the top left: A4-1, A4-3, A4-4, A4-2)

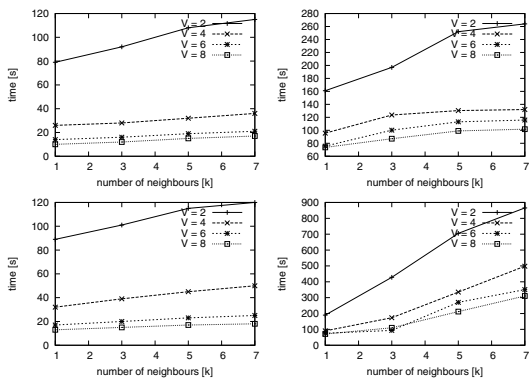


Fig. 2. Time complexity for the Adult dataset (clockwise, starting from the top left: A4-1, A4-3, A4-4, A4-2)

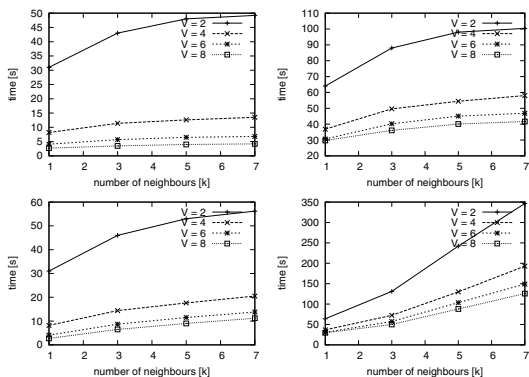


Fig. 3. Time complexity for the Letter Recognition dataset (clockwise, starting from the top left: A4-1, A4-3, A4-4, A4-2)

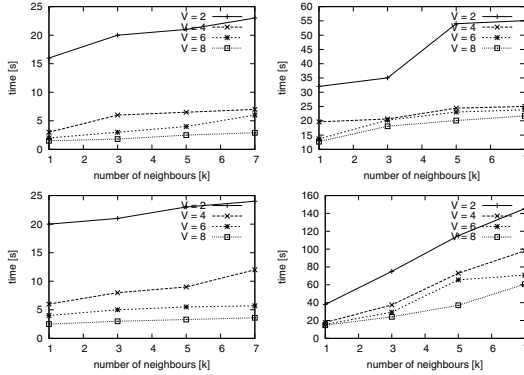


Fig. 4. Time complexity for the Splice-junction Gene Sequences dataset (clockwise, starting from the top left : A4-1, A4-3, A4-4, A4-2)

- With the increase of the privacy level the computational complexity of the introduced methods is also rising. However, differences between the methods A4-1 and A4-2 are practically unnoticeable, but A4-2 provides higher privacy at almost none of the additional computational expense.
- On the other hand the difference between the privacy global data privacy (A4-2) and local view privacy (A4-3, A4-4) approaches is very significant. The computational cost rise is several times greater.
- Algorithms A4-3 and A4-4 offer privacy at highest levels (local view privacy). A4-4 is almost four times slower than A4-3. Yet this can be explained by the complex approach to the databases query, offered by A4-4. In the exchange for longer computational time it increases the chance to reveal the smallest possible number of objects by each of the sides.
- For A4-1, A4-2 and A4-3 the computational time depends very little on k parameter for a small number of features, but for Arcene database, with high dimensionality feature space (10000), k parameter is highly correlated with the execution time. For A4-4 algorithm this dependency is always very strong.

The results of experiments did not surprise us, because we expected that higher privacy level requires higher computational cost, but it is worth noting that the additional computational cost is lower than usually reported. Due to the minimal difference in time complexity between the algorithms A4-1 and A4-2 it is recommended to always use the second proposed method, because it offers higher privacy at almost no additional cost. In case of choosing local view privacy level the choice should depend on the application. If the user is concerned about the execution time, then A4-3 method is recommended. It offers a high level of the privacy with about two-three times less computational cost and is very weakly correlated to the size of k parameter. If the computational complexity is not an issue and maximum privacy is required the method A4-4 seems to be the proper choice.

5 Conclusions

The paper dealt with a problem of privacy preserving for the classification tasks. We proposed the new definition of privacy and the original taxonomy of privacy preserving algorithms. Splitting the idea of privacy into 5 levels allows us to introduce a flexible framework that can be adjusted to personal needs and offers balance between safety and computational costs. We discussed four modifications of k -NN algorithm which take the mentioned above privacy taxonomy into consideration. On the basis of experimental results we formulated recommendations for practical implementations of proposed methods and we showed that a good level of security can be achieved without using additional time consuming algorithms. Introducing modifications which embedded the privacy-preserving task into the nature of minimal distance classification allowed to present fast and efficient tasks for such problems. We believe that the proposed concept can be useful during a real project of a distributed computer recognition system which could use partitioned datasets and should respect privacy aspects of data.

Acknowledgement. The work was supported by the statutory funds of the Department of Systems and Computer Networks, Wrocław University of Technology and by The Polish National Science Centre under the grant N N519 576638 which is being realized in years 2010-2013.

References

1. Alpaydin, E.: Introduction to Machine Learning, 2nd edn. The MIT Press, London (2010)
2. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
3. Chitti, S., Liu, L., Xiong, L.: Mining Multiple Private Databases using Privacy Preserving kNN Classifier, Technical Reports TR-2006-008, Emory University (2006)
4. Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., Zhu, M.Y.: Tools for privacy preserving data mining. SIGKDD Explorations, 28–34 (2002)
5. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Trans. on Inform. Theory 13(1), 21–27 (1967)
6. Devroye, L.: On the inequality of cover and hart in nearest neighbor discrimination. IEEE Trans. on Pat. Anal. and Mach. Intel. 3, 75–78 (1981)
7. Han, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publ. Inc., San Francisco (2005)
8. Lindell, Y., Pinkas, B.: Privacy Preserving Data Mining. Journal of Cryptology 15(3), 177–206 (2004)
9. Lindell, Y., Pinkas, B.: A proof of security of yao’s protocol for two-party computation. Journal of Cryptology 22(2), 161–188 (2009)
10. Moor, J.H.: The future of computer ethics: You ain’t seen nothin’ yet! Ethics and Information Technology 3, 89–91 (2001)
11. Nissenbaum, H.: Can we Protect Privacy in Public? In: Computer Ethics Philosophical Enquiry ACM/SIGCAS Conference, Rotterdam, The Netherlands (1997)
12. Westin, A.F.: Privacy and Freedom. The Bodley Head Ltd. (1970)

Borderline Kernel Based Over-Sampling

María Pérez-Ortiz, Pedro Antonio Gutiérrez, and César Hervás-Martínez*

University of Córdoba, Dept. of Computer Science and Numerical Analysis
Rabanales Campus, Albert Einstein building, 14071 - Córdoba, Spain
{i82perom,pagutierrez,chervas}@uco.es

Abstract. Nowadays, the imbalanced nature of some real-world data is receiving a lot of attention from the pattern recognition and machine learning communities in both theoretical and practical aspects, giving rise to different promising approaches to handling it. However, preprocessing methods operate in the original input space, presenting distortions when combined with kernel classifiers, that operate in the feature space induced by a kernel function. This paper explores the notion of empirical feature space (a Euclidean space which is isomorphic to the feature space and therefore preserves its structure) to derive a kernel-based synthetic over-sampling technique based on borderline instances which are considered as crucial for establishing the decision boundary. Therefore, the proposed methodology would maintain the main properties of the kernel mapping while reinforcing the decision boundaries induced by a kernel machine. The results show that the proposed method achieves better results than the same borderline over-sampling method applied in the original input space.

1 Introduction

Imbalanced classification is one of the current challenges for machine learning [1,2], since it has been shown to hinder the learning performance of classification algorithms. Imbalanced classification problems are very common in many real-world domains, such as medical diagnosis, text categorization, fraud detection or information retrieval, contexts where usually the minority class happens to be more interesting than the majority one, but also more difficult to model due to the low number of available patterns. Since most traditional learning systems have been designed to work on balanced data, they will usually be focused on improving overall performance and be biased towards the majority class, consequently harming the minority one [3]. To cope with this issue, several algorithms have been designed over the years to over-sample minority samples and to under-sample the majority ones, the Synthetic Minority Over-sampling Technique [1] (SMOTE) being one of the most representatives for the first group, among others.

* This work has been partially subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P11-TIC-7508 project of the “Junta de Andalucía” (Spain).

In the context of kernel classifiers [4], since the very first introduction of the support vector machine paradigm (Support Vector Classifier or SVC), we have witnessed a huge development in theory and methodologies of what is known as kernel-based methods: advances in performance theory, different variants of kernel classifiers and regressors, algorithms for feature selection and extraction, all that accompanied by countless successful applications. Moreover, the success of kernel methods can be attributed to the joint use of a robust classification procedure (such as the large margin hyperplane principle) and a convenient and versatile way of preprocessing the patterns (the kernel trick). However, very little has been done in the context of imbalanced classification, and more specifically, concerning over-sampling in the feature space. This is essentially the main aim of this paper because when these classifiers are combined with other preprocessing techniques which operate in the input space, some obvious distortions are found, given that they operate in different spaces. The ideal approach would be preprocess the training patterns in the feature space, although this is not possible since the only information available is the dot products of their images. To deal with this issue, this paper makes use of the notion of empirical feature space [5,6], which preserves the geometrical structure of the original feature space, given that distances and angles in the feature space are uniquely determined by dot products and that the dot products of the corresponding images are the original kernel values. This empirical feature space is Euclidean, so it provides a tractable framework to study the spatial distribution of the mapping function $\Phi(\cdot)$ [7], to measure class separability [6] and to optimize the kernel [6,8]. Besides, the notion of empirical kernel feature space has been used for the kernelization of all kinds of linear classifiers [9,10], with the advantage that the algorithm does not need to be formulated to deal with dot products.

Therefore, the main aim of this paper is to check whether the empirical feature space provides a more suitable space than the input space for performing over-sampling. This Euclidean space is isomorphic to the feature space, hence we hypothesize that the synthetic patterns generated will be better adapted to the kernel machine classifier. Borderline over-sampling [11] has been chosen for the experimentation since we consider that borderline examples are more informative for a large margin based classifier such as SVM (this borderline area is more crucial for establishing the decision boundary) and also most prone to be misclassified. Indeed, performing over-sampling on this area has been demonstrated to make more benefit than performing it on the whole minority class [11,12]. For this purpose, an efficient way of selecting informative instances from the pool of samples is also needed, this step being usually computed in the input space, rather than in the feature one, which is also one of the hypotheses of the paper: that, for a kernel machine, borderline patterns will be better chosen in the feature space than in the input space, given that the kernel machine operates in this feature space.

The idea of over-sampling in the feature space have been also researched in [13], where synthetic instances were generated by using the geometric interpretation of the dot products in the kernel matrix, and the pre-images of these

synthetic instances were approximated based on a distance relation between the feature space and the input one, since inverse mapping $\Phi(\cdot)^{-1}$ from the feature space to input space is not available. Finally, the approximation of these pre-images are appended to the original dataset to train a SVM. Note that in our case, the over-sampling is performed in the empirical feature space, thus our methodology is free of the computational cost and assumptions of this inverse mapping approximation.

The paper is organized as follows: Section 2 shows a description of the methodology used; Section 3 describes the experimental study and analyses the results obtained; and finally, Section 4 outlines some conclusions.

2 Methodology

The goal in binary classification could be said to assign an input vector $\mathbf{x} \in \mathbb{R}^d$ to one of the classes \mathcal{C}_+ and \mathcal{C}_- (corresponding this labelling to the output space \mathcal{Y}). The objective is to find a prediction rule $f : \mathcal{X} \rightarrow \mathcal{Y}$ by using an i.i.d. training sample $D = \{\mathbf{x}_i, y_i\}_{i=1}^m \in \mathcal{X} \times \mathcal{Y}$. The methodology here proposed is based on performing over-sampling in the empirical feature space using the patterns on the boundary of the minority class. Consequently, the notion of empirical feature space is firstly described. Then, we describe how to extend the borderline SMOTE algorithm to better handle imbalanced datasets when applying kernel classifiers.

2.1 Empirical Feature Space

In this section, the empirical feature space spanned by the training data is defined. Let \mathcal{H} denote a high-dimensional or infinite-dimensional Hilbert space. Then, for any mapping of patterns $\Phi : \mathcal{X} \rightarrow \mathcal{H}$, the inner product $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathcal{H}}$ of the mapped inputs is known as a kernel function, giving rise to a symmetric and positive semidefinite matrix (known as Gram or kernel matrix \mathbf{K}) from a given input set \mathcal{X} . By definition, these matrices can be diagonalised as follows:

$$\mathbf{K}_{(m \times m)} = \mathbf{P}_{(m \times r)} \cdot \mathbf{M}_{(r \times r)} \cdot \mathbf{P}_{(r \times m)}^T, \quad (1)$$

where $(\cdot)^T$ is the transpose operation, \mathbf{M} is a diagonal matrix containing the r positive eigenvalues of \mathbf{K} in decreasing order, and \mathbf{P} consists of the eigenvectors associated to those r eigenvalues. The empirical feature space is a Euclidean space preserving the dot product information about \mathcal{H} contained in \mathbf{K} . The mapping from the input space to a r -dimensional empirical feature space can be defined as $\Phi_r^e : \mathcal{X} \rightarrow \mathbb{R}^r$, where r is the rank of \mathbf{K} . This space is isomorphic to the embedded feature space \mathcal{H} , but presents all the advantages of being Euclidean:

$$\Phi_r^e : \mathbf{x}_i \rightarrow \mathbf{M}^{-1/2} \cdot \mathbf{P}^T \cdot (\mathcal{K}(\mathbf{x}_i, \mathbf{x}_1), \dots, \mathcal{K}(\mathbf{x}_i, \mathbf{x}_m))^T. \quad (2)$$

It is easy to check that the kernel matrix of the training images obtained by this transformation is \mathbf{K} , when considering the standard dot product [5,6]. Note that this transformation corresponds to the principal component analysis *whitening* step [14], although applied to the kernel matrix, instead of the covariance matrix. Although the whole set of all r positive eigenvalues could be considered, a smaller set (in this case, for simplicity, a 10-dimensional set) has been chosen in this paper by choosing the p dominant eigenvalues and their associated eigenvectors. The choice of this smaller set limits the dimensionality of the empirical feature space and make more robust the process of over-sampling by simplifying the space, given the concentration of spectral measures.

Fig. 1 shows the case of a synthetic dataset concerning a non-linearly separable classification task and its transformation to the two-dimensional empirical feature space induced by the well-known standard Gaussian kernel, which is linearly separable.

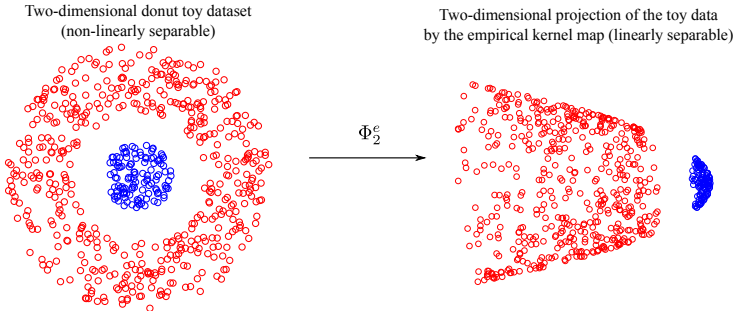


Fig. 1. Synthetic two-dimensional dataset representing a non-linearly separable classification problem and its transformation to the 2 dominant dimensions of the empirical feature space induced by the Gaussian kernel function (linearly separable problem)

2.2 Borderline Over-Sampling in the Empirical Feature Space

The main idea for the proposed method is to use the empirical feature space to apply preprocessing algorithms, because preprocessed patterns would better suit the kernel machine classifier later considered. In this paper, the borderline SMOTE algorithm was selected to decrease the problems caused by imbalanced datasets when applying a kernel classifier.

Borderline over-sampling [11] is based on the idea of generating new synthetic patterns on the borderline between different classes, as these patterns are considered as being more probable to be misclassified. Thus, the first step corresponds to the identification of these patterns that are “in danger” of being misclassified, which is usually done by examining the neighborhood of the pattern considered, e.g. if all the nearest neighbors correspond to the minority class, the pattern is not considered as a borderline example, however, if half of the nearest neighbors belong to the minority class and the other half to the majority one, the pattern can be considered as a borderline one. Finally, borderline examples are the ones

considered for generating new synthetic patterns by means of the well-known SMOTE technique [1]. Therefore, when considering the empirical feature space rather than the original input one, not only the process of generating new examples change as the space used is different, but also the patterns chosen as borderline.

Concerning the proposed method, first of all, the empirical feature space induced by a kernel function \mathcal{K} in the training set is computed. Formally, $\mathbf{T}^e_{(m \times r)}$ is the matrix generated by applying the empirical kernel map Φ_r^e (see (2)) to the training patterns. Then, the standard borderline SMOTE algorithm [11] is applied over the class images of this \mathbf{T}^e matrix, resulting in the generation of n new synthetic images of patterns, arranged in the matrix $\mathbf{S}^e_{(n \times r)}$ (note that all these new patterns will belong to the minority class). The new synthetic examples will be used to complete the kernel matrix, obtaining their dot product with respect to the rest of training patterns, i.e. $\mathbf{KS}^e_{i,j} = \mathbf{T}^e_i \cdot \mathbf{S}^e_j, 1 \leq i \leq m, 1 \leq j \leq n$, and with respect to themselves $\mathbf{SS}^e_{i,j} = \mathbf{S}^e_i \cdot \mathbf{S}^e_j, 1 \leq i, j \leq n$, where \mathbf{T}^e_i is the empirical space representation of the i -th training pattern, and \mathbf{S}^e_i is the i -th synthetic sample previously generated. Using these matrices, the over-sampled training Gram matrix \mathbf{K}^* will be composed as follows:

$$\mathbf{K}^*_{(m+n) \times (m+n)} = \begin{pmatrix} \mathbf{K}_{(m \times m)} & \mathbf{KS}^e_{(m \times n)} \\ \left(\mathbf{KS}^e_{(m \times n)}\right)^T & \mathbf{SS}_{(n \times n)} \end{pmatrix}, \quad (3)$$

where \mathbf{K} is the original kernel matrix. For the generalization phase, the same steps are considered to complete the test kernel matrix, taking into account that the empirical feature space images of the test patterns are derived using the same Φ_r^e transformation (considering only the training data). Fig. 2 shows the main steps of the proposed algorithm: Borderline Kernel SMOTE (BKS).

Algorithm BKS

- **Input:** Training patterns (\mathbf{Tr}) and training targets (\mathbf{Trg}).
- **Output:** Over-sampled training kernel matrix (\mathbf{K}^*).
 1. Compute kernel matrix \mathbf{K} for training patterns.
 2. Compute the empirical kernel map Φ_r^e via \mathbf{K} .
 3. Map training patterns to the empirical feature space using Φ_r^e (\mathbf{T}^e).
 4. Apply borderline SMOTE with the new representation \mathbf{T}^e of the training patterns and obtain a new set \mathbf{S}^e of synthetic data.
 5. Complete the over-sampled kernel matrix \mathbf{K}^* with the new synthetic patterns and their dot product according to (3).

Fig. 2. Different steps for the kernel over-sampling algorithm proposed

Given that the over-sampling technique operates in r dimensions (kernel matrix rank), instead of d (dimensionality of the input space), what is noteworthy is the applicability of the proposed method to bioinformatics datasets where

the number of features tend to be much higher than the number of samples ($r \ll d$), and where imbalanced datasets are commonly found. Additionally, as an advantage of the method, there is no need to treat the data attributes differently (taking into account their nature) since all of them are real, unlike in the original SMOTE.

As a final remark, in order to clarify the usefulness of performing the over-sampling in the feature space, let us analyze the case presented in Fig. 3, where a toy non-linearly separable dataset has been represented. The top part of the figure corresponds to the synthetic dataset created and its transformation via the empirical kernel map, while the bottom part includes information about the 5-nearest neighbors for each pattern. From this figure, one can appreciate that despite the fact that k -nearest neighbors is a nonlinear methodology, it is very sensitive to the correct choice of k , in such a way that we could be generating new synthetic patterns in an inappropriate region (as the bottom left plot where the over-sampling is generated in the input space). However, if we consider the empirical feature space instead (as in the right part of the figure), the over-sampling is less sensitive to the choice of k , since, in this space, the separation between the patterns is easier (ideally, linearly separable), which is one of the main characteristics of the kernel trick. Note that the representation of the empirical feature space plotted in the right part of the figure is only a two-dimensional approximation, thus we are obviating useful information.

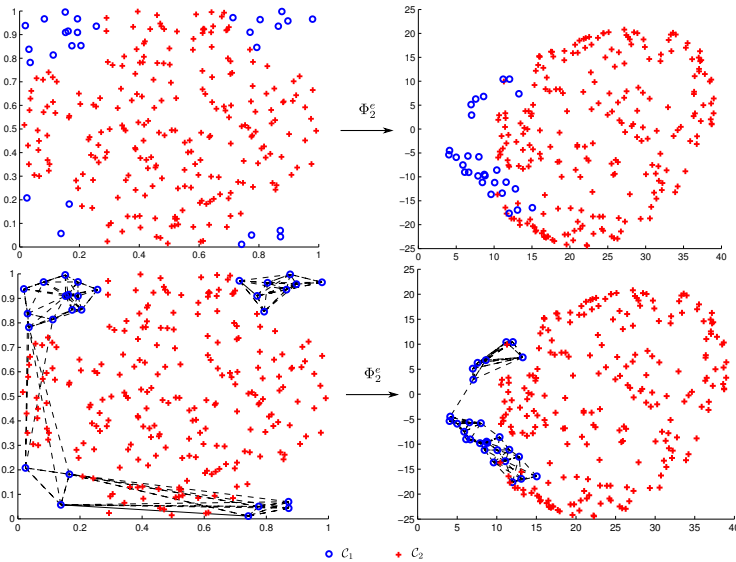


Fig. 3. Toy two-dimensional non-linearly separable dataset and the transformation to the 2 dominant dimensions of the empirical feature space induced by the Gaussian kernel function. Dashed lines represent the 5-nearest neighbors of each pattern belonging to the minority class.

Table 1. Characteristics of the benchmark datasets used for the experimentation (number of patterns, features and imbalance ratio (IR))

Dataset	Patterns	Features	IR
liver	345	6	1.38
bands	365	19	1.70
vehicle1	846	18	2.90
ecoli1	336	7	3.36
ecoli2	336	7	5.46
glass6	214	9	6.38
yeast0359-78	506	8	9.12
vowel0	988	13	9.98
yeast1-7	459	8	14.30
yeast1289-7	947	8	30.57

All nominal variables are transformed into binary ones

3 Experimental Results

The proposed method has been tested considering the Support Vector Classifier (SVC) [15] and the well-known borderline SMOTE [11]. Our methodology (Borderline Kernel SMOTE, BKS) is compared to the original borderline SMOTE in the input space (BS), and to the results without over-sampling. 10 binary benchmark datasets from the UCI repository [16] with different imbalance ratios (proportion of majority patterns with respect to minority ones) have been tested to analyze the performance of the methods in different situations. The characteristics of these datasets can be seen in Table 1. As done in other over-sampling state-of-the-art works [3], some multiclass datasets have also been considered by grouping some classes, e.g. ecoli1 represents the ecoli dataset when considering class 1 versus the rest, and yeast0359-78 is the yeast dataset when grouping classes 0, 3, 5, and 9 versus classes 7 and 8 in order to obtain higher imbalance ratio values.

A stratified 5-fold technique was performed to divide the data and the results are taken as mean and standard deviation of the selected measures. The Gaussian kernel was used. The kernel width and the cost parameter of SVC was selected within the values $\{10^{-3}, 10^{-2}, \dots, 10^3\}$, by means of a nested 5-fold method applied to the training set. The number of synthetic patterns generated was that needed to balance the distributions, i.e. after applying the over-sampling process, the number of majority and minority patterns were the same.

The results have been reported in terms of three metrics, two of them specially designed to deal with imbalanced datasets:

1. The well-known Accuracy metric (Acc), which corresponds to the ratio of correctly classified patterns and measures overall performance.
2. The Geometric Mean of the sensitivities ($GM = \sqrt{S_p \cdot S_n} \cdot 100$), where S_p is the sensitivity for the positive class (ratio of correctly classified patterns considering only this class) and S_n is the sensitivity for the negative one.
3. The Minimum Sensitivity [17] ($MS = \min\{S_p, S_n\} \cdot 100$), which can be defined as the minimum value of the sensitivities for each class.

Table 2. Results achieved by the three methods considered for the different metrics

Dataset	Algorithm	Acc(%)	GM(%)	MS(%)
liver	SVC	71.03 ± 8.05	68.28 ± 8.59	58.57 ± 10.41
	BS+SVC	69.86 ± 6.68	68.27 ± 6.22	61.88 ± 6.43
	BKS+SVC	71.30 ± 9.28	70.06 ± 9.45	64.21 ± 10.62
bands	SVC	71.76 ± 4.61	66.46 ± 8.15	55.49 ± 14.17
	BS+SVC	71.49 ± 5.50	65.95 ± 10.25	55.33 ± 16.50
	BKS+SVC	70.11 ± 6.85	68.63 ± 9.53	61.56 ± 12.94
vehicle1	SVC	85.34 ± 4.13	80.35 ± 7.85	72.32 ± 12.68
	BS+SVC	86.05 ± 1.72	83.48 ± 1.70	78.58 ± 3.99
	BKS+SVC	83.10 ± 2.09	84.48 ± 2.52	81.55 ± 1.94
ecoli1	SVC	90.20 ± 4.94	85.38 ± 5.82	77.75 ± 8.95
	BS+SVC	87.52 ± 4.08	84.52 ± 6.75	77.15 ± 11.51
	BKS+SVC	90.18 ± 2.92	86.45 ± 3.76	80.38 ± 7.45
ecoli2	SVC	94.95 ± 2.23	90.55 ± 3.07	84.73 ± 4.87
	BS+SVC	94.94 ± 2.26	93.03 ± 5.22	89.14 ± 7.98
	BKS+SVC	97.02 ± 2.11	95.11 ± 3.98	91.84 ± 6.94
glass6	SVC	95.32 ± 2.88	85.73 ± 9.39	75.33 ± 16.26
	BS+SVC	93.44 ± 5.58	86.33 ± 12.02	78.13 ± 18.96
	BKS+SVC	95.82 ± 6.31	92.48 ± 14.52	87.78 ± 21.88
yeast0359-78	SVC	87.54 ± 5.81	50.88 ± 13.05	30.00 ± 15.81
	BS+SVC	79.05 ± 3.56	64.18 ± 11.21	49.38 ± 15.99
	BKS+SVC	70.93 ± 10.35	66.72 ± 7.24	57.74 ± 11.86
vowel0	SVC	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
	BS+SVC	99.90 ± 0.23	99.94 ± 0.12	99.89 ± 0.25
	BKS+SVC	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
yeast1-7	SVC	94.12 ± 1.81	48.30 ± 27.42	30.00 ± 18.26
	BS+SVC	84.99 ± 5.44	69.14 ± 26.45	59.61 ± 24.73
	BKS+SVC	81.94 ± 7.44	77.07 ± 12.66	67.98 ± 14.47
yeast1289-7	SVC	97.25 ± 0.69	45.30 ± 27.43	26.67 ± 19.00
	BS+SVC	81.62 ± 5.60	63.55 ± 16.73	51.53 ± 25.12
	BKS+SVC	79.39 ± 7.89	69.71 ± 10.66	60.60 ± 18.80

The best method is in **bold** face and the second one in *italics*

Table 3. Mean and ranking values obtained for each methodology and measure

Measure	SVC		BS+SVC		BKS+SVC	
	Mean	Rank	Mean	Rank	Mean	Rank
<i>Acc</i>	88.75	1.45	<i>84.88</i>	2.4	83.98	<i>2.15</i>
<i>GM</i>	72.12	2.55	<i>77.84</i>	2.4	81.07	1.05
<i>MS</i>	61.08	2.65	<i>70.06</i>	2.3	75.36	1.05

The best method is in **bold** face and the second one in *italics*

The measure considered during the hyperparameter selection was *GM*, given its robustness for imbalanced datasets. All the test results of these experiments can be seen in Table 2 and the mean and rankings of these results in Table 3.

From the results obtained, several conclusions can be drawn. Firstly, the good performance of the proposed method can be seen analyzing *GM* and *MS* measures, where it can be seen that the application of the over-sampling technique in the empirical feature space outperforms the results achieved when applying it in the original input space. Indeed, the ranking of these measures for the SVC and BS+SVC algorithms are similar, indicating that the use of an over-sampling technique in the original input space may not incorporate enough useful information for a kernel machine. Furthermore, although standard deviations corresponding to *GM* and *MS* are high, due to the drastic nature of these measures, in most

of the cases, standard deviations of BKS are lower than the ones associated with BS. Concerning *Acc*, the proposed method achieves comparable results to those obtained by the other methods (especially for low IR values). However, one can appreciate that in some cases deteriorating the classification of the majority class (and therefore the overall performance) is needed in order to classify correctly the minority one (this is the case of the datasets yeast0359-78, yeast1-7 and yeast1289-7). With concern to very low IR values (the case of liver and bands datasets), the over-sampling proposed algorithm do not deteriorate the SVC solution and is even able to obtain better values for *GM* and *MS*. Finally, for the vowel0 dataset, it can be seen that the application of BS is not successful, since the original SVC obtains an optimal solution that is not found when performing the over-sampling in the input space. However, when performing the over-sampling in the feature space induced by the kernel, the performance of the classifier is not deteriorated.

To quantify whether a statistical difference exists among the algorithms compared, the non-parametric Friedman's test [18] (with $\alpha = 0.05$) has been applied to the mean rankings for the three measures considered, rejecting the null-hypothesis that all algorithms perform similarly for *GM* and *MS*, and accepting it for *Acc*. The confidence interval was $C_0 = (0, F_{(\alpha=0.05)} = 3.55)$, and the corresponding F-value was $2.88 \in C_0$, $19.35 \notin C_0$ and $21.77 \notin C_0$ for *Acc*, *GM* and *MS*, respectively. Furthermore, the Nemenyi test has also been applied concluding that there are statistically significant differences for $\alpha = 0.05$ in *GM* and *MS* (the Nemenyi critical difference being 1.04782) when comparing BKS+SVC with SVC (with ranking differences of 1.5 and 1.6, respectively) and with BS+SVC (with ranking differences of 1.35 and 1.25, respectively).

4 Conclusions and Future Work

This paper explores the idea of performing over-sampling in the class boundary of the empirical feature space related to a kernel function. We focus on the imbalanced binary classification paradigm, and the proposed method has been tested with the standard Support Vector Classifier and the borderline SMOTE algorithm, achieving better results than when applying the same preprocessing in the original input space, specially for metrics designed for imbalanced classification. As future work, the performance of different kernel functions for performing kernel over-sampling could be studied to analyze the kernel function to use according to the nature of the data. Furthermore, in the same vein as this paper, an analytical methodology [19] could be used to compute the number of relevant dimensions for the empirical feature space (note that in our case this value was prefixed for the sake of simplicity).

References

1. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16, 321–357 (2002)

2. Tang, Y., Zhang, Y.Q., Chawla, N.V., Krasser, S.: SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* 39(1), 281–288 (2009)
3. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 42(4), 463–484 (2012)
4. Schölkopf, B., Smola, A.J.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press (2001)
5. Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Müller, K.R., Rätsch, G., Smola, A.J.: Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks* 10, 1000–1017 (1999)
6. Xiong, H., Swamy, M.N.S., Ahmad, M.O.: Optimizing the kernel in the empirical feature space. *IEEE Transactions on Neural Networks* 16(2), 460–474 (2005)
7. Yan, F., Mikolajczyk, K., Kittler, J., Tahir, M.A.: Combining multiple kernels by augmenting the kernel matrix. In: El Gayar, N., Kittler, J., Roli, F. (eds.) *MCS 2010*. LNCS, vol. 5997, pp. 175–184. Springer, Heidelberg (2010)
8. Xiong, H., Swamy, M.N.S., Ahmad, M.O.: Learning with the optimized data-dependent kernel. In: *Proc. of the 2004 Conference on Computer Vision and Pattern Recognition Workshop, CVPRW*, vol. 6, pp. 95–101. IEEE Computer Society (2004)
9. Abe, S., Onishi, K.: Sparse least squares support vector regressors trained in the reduced empirical feature space. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) *ICANN 2007*. LNCS, vol. 4669, pp. 527–536. Springer, Heidelberg (2007)
10. Xiong, H.: A unified framework for kernelization: The empirical kernel feature space. In: *Chinese Conference on Pattern Recognition, CCPR*, pp. 1–5 (November 2009)
11. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) *ICIC 2005*. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005)
12. Wang, H.Y.: Combination approach of smote and biased-svm for imbalanced datasets (2008)
13. Zeng, Z.-Q., Gao, J.: Improving SVM classification with imbalance data set. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) *ICONIP 2009, Part I*. LNCS, vol. 5863, pp. 389–398. Springer, Heidelberg (2009)
14. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10(5), 460–474 (1998)
15. Cortes, C., Vapnik, V.: Support-vector networks. *Machine Learning* 20(3), 273–297 (1995)
16. Asuncion, A., Newman, D.: *UCI machine learning repository* (2007)
17. Fernández-Caballero, J.C., Martínez-Estudillo, F.J., Hervás-Martínez, C., Gutiérrez, P.A.: Sensitivity versus accuracy in multiclass problems using memetic pareto evolutionary neural networks. *IEEE Transactions on Neural Networks* 21(5), 750–770 (2010)
18. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
19. Braun, M.L., Buhmann, J.M., Müller, K.R.: On relevant dimensions in kernel feature spaces. *J. Mach. Learn. Res.* 9, 1875–1908 (2008)

Discrimination of Resting-State fMRI for Schizophrenia Patients with Lattice Computing Based Features

Darya Chyzhyk and Manuel Graña

Computational Intelligence Group
Dept. CCIA, UPV/EHU, Apdo. 649, 20080 San Sebastian, Spain
www.ehu.es/ccwintco

Abstract. Resting state fMRI data can be used to find biomarkers of specific neurological conditions, such as schizophrenia. In this paper we report results on the discrimination between schizophrenia patients and healthy control, as well as the discrimination of subpopulations of schizophrenia patients with and without auditory hallucinations. Data features for classification are obtained as follows: a Multivariate reduced ordering based on a h -function constructed from Lattice Autoassociative Memories recall. The Pearson correlation coefficient between the h -function values and the categorical variable at each voxel site allows to identify the most informative voxel sites. Feature vectors are constructed as the h -function values at these sites. Results on a database of healthy controls and schizophrenia patients with and without auditory hallucinations show that the approach can provide accurate discrimination between these populations.

1 Introduction

Correlation of low frequency oscillations in diverse areas of the brain in resting state fMRI data has been used to study the connectivity of brain activations [1,2,3], uncovering a kind of brain functional fingerprint. One strong reason for resting state fMRI experiments is that they do not impose constraints on the cognitive abilities of the subjects. Computational approaches applied include hierarchical clustering [4], independent component analysis (ICA) [5,6,7], fractional amplitude of low frequency analysis [8], multivariate pattern analysis (MVPA) [9,10]. Resting state fMRI has being found useful for performing studies on brain evolution based on the variations in activity of the default mode network [9], depression (using regional homogeneity measures) [11], Alzheimer's Disease [12], and schizophrenia.

Schizophrenia is a severe psychiatric disease that is characterized by delusions and hallucinations, loss of emotion and disrupted thinking. Functional disconnection between brain regions is suspected to cause these symptoms, because of known aberrant effects on gray and white matter in brain regions that overlap with the default mode network. Resting state fMRI studies [13,14,15] have indicated aberrant default mode functional connectivity in schizophrenic patients.

Resting state studies for schizophrenia patients with auditory hallucinations have also been performed [16] showing reduced connectivity. Recent findings [17] show effects on the resting state network localizations correlated with voxels in the left Heschl's gyrus (LHG; MNI coordinates -42,-26,10) from the auditory cortex effect related to the auditory hallucinations in schizophrenic patients.

Definition of morphological operators in multivariate images needs the definition of appropriate orders in the vector space. Previous works [18,19] have applied a Lattice Computing [20,21,22] supervised h -ordering based on Lattice Auto-Associative Memories (LAAMs) [23,24], *LAAM-supervised ordering*, to the definition of morphological operators. These works have found that it is possible to find by group analysis differences between populations of schizophrenia patients, with and without auditory hallucinations, and healthy controls. The approach is a mathematical morphology correspondent to the correlation [25] based approaches to the analysis of resting state fMRI searching for networks of low frequency synchronized components in the brain. Here the h -function is the similarity measure used to perform brain network identification. In this paper we report results on the discrimination of individuals based on the feature vectors extracted from voxel sites of the h -function map related to the left Heschl's gyrus. Classification with k-NN classifiers provides baseline results that are quite encouraging.

2 Methods

2.1 Multivariate Ordering

One way to accomplish a Multivariate Mathematical Morphology is through the definition of a reduced ordering [26]. A h -ordering is defined by a surjective mapping of the original data set onto a complete lattice $h : X \rightarrow \mathbb{L}$, so that the order in the target lattice induces a total order on the original data set X , that is:

$$\mathbf{x} \leq_h \mathbf{y} \Leftrightarrow h(\mathbf{x}) \leq h(\mathbf{y}); \forall \mathbf{x}, \mathbf{y} \in X. \quad (1)$$

The reduced ordering can be defined on the basis of a supervised classifier trained with some pixel values extracted from the image. Often, two class discrimination between foreground and background classes is considered.

2.2 LAAM's h -Mapping

The LAAM h -mapping is defined as the Chebyshev distance between the original pattern vector and the recall obtained from the LAAM. Formally, given a sample data vector $\mathbf{x} \in \mathbb{R}^n$ and a non-empty training set $X = \{\mathbf{x}_i\}_{i=1}^K$, $\mathbf{x}_i \in \mathbb{R}^n$ for all $i = 1, \dots, K$, the LAAM h -mapping is given by:

$$h_X(\mathbf{c}) = d_C(\mathbf{x}^\#, \mathbf{x}), \quad (2)$$

where $\mathbf{x}^\# \in \mathbb{R}^n$ is the recalling response of dilative LAAM M_{XX} to the input of vector \mathbf{x} , i.e. $\mathbf{x}_M^\# = M_{XX} \boxtimes \mathbf{x}$. The erosive memory W_{XX} recall, i.e. $\mathbf{x}_W^\# = W_{XX} \boxtimes \mathbf{x}$, could be used alternatively. Function $d_C(\mathbf{a}, \mathbf{b})$ denotes the Chebyshev

distance between two vectors, given by the greatest absolute difference between the vectors' components: $d_C(\mathbf{a}, \mathbf{b}) = \bigvee_{i=1}^n |a_i - b_i|$. Here \bigvee and \bigwedge denote the max and min matrix product [27,24], respectively defined as follows:

$$C = A \bigvee B = [c_{ij}] \Leftrightarrow c_{ij} = \bigvee_{k=1, \dots, n} \{a_{ik} + b_{kj}\},$$

$$C = A \bigwedge B = [c_{ij}] \Leftrightarrow c_{ij} = \bigwedge_{k=1, \dots, n} \{a_{ik} + b_{kj}\}.$$

2.3 Background/Foreground LAAM h -Supervised Orderings

The Background/Foreground (B/F) LAAM h -supervised ordering is constructed applying Foreground LAAM h -mapping of Eq. (2) to disjoint background B and foreground F training sets, obtaining mappings h_B and h_F , respectively. We define a Background/Foreground (B/F) LAAM h -mapping $h_r(\mathbf{x})$ combining both h_B and h_F into an h -mapping as follows:

$$h_r(\mathbf{x}) = h_F(\mathbf{x}) - h_B(\mathbf{x}), \quad (3)$$

which is positive for $\mathbf{x} \in \mathcal{F}(B)$, and negative for $\mathbf{x} \in \mathcal{F}(F)$. Therefore, we assume it as a discriminant function such that $h_r(\mathbf{x}) > 0$ corresponds to pixels in the background class, and $h_r(\mathbf{x}) < 0$ to pixels in the foreground class. Points where $h_r(\mathbf{x}) = 0$ holds correspond to the decision boundary.

Computing the B/F LAAM h -function produces a real valued map over the brain volume, where functional networks are identified applying a threshold to this map.

2.4 Pearson Correlation

The Pearson correlation coefficient is given by:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \quad (4)$$

where $r \in [-1, 1]$, $r = 1$ means that two variables have perfect positive correlation and $r = -1$ means that there is a perfect negative correlation between them. In our case Pearson correlation evaluates the nexus between *a priori* known class labels and fMRI neural connectivity by means of h -function.

2.5 Experimental Pipeline

Figure 1 shows the graphical description of our experimental process. Resting state fMRI data is first preprocessed to ensure that all fMRI volumes are aligned and warped to the spatially normalized structural T1-weighted data. On the normalized data, we compute the B/F LAAM h -mapping where the Background

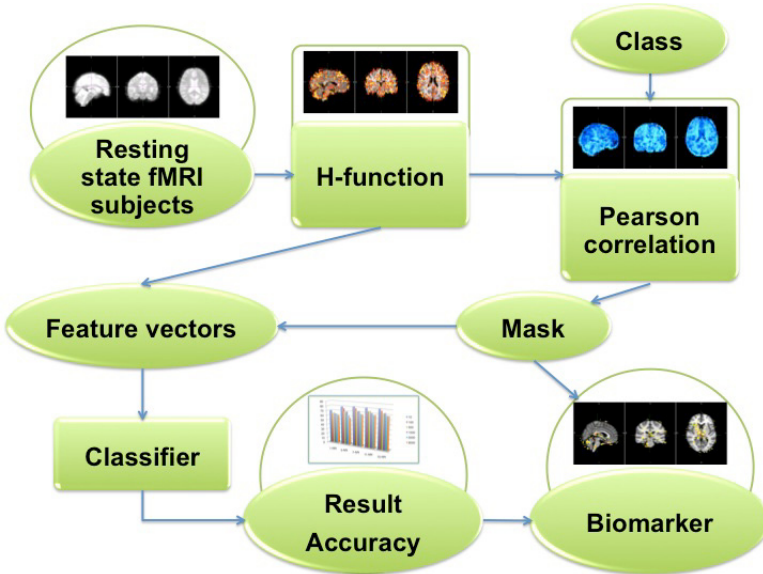
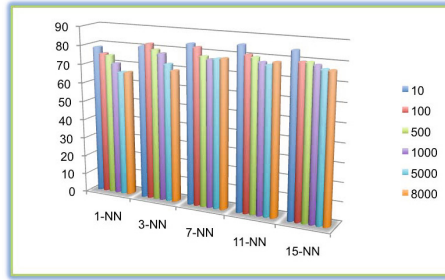


Fig. 1. Pipeline of our experimental design

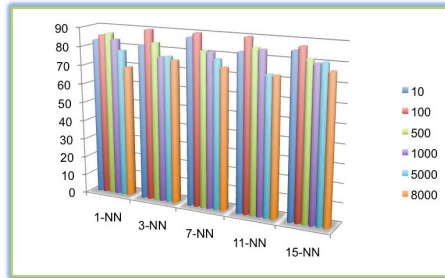
data corresponds to Cerebrospinal fluid (CSF) in the brain ventricle voxels, and the Foreground data is a selection of voxels in the LHG according to [17]. The h -mappings are used to compute across volume voxel-wise Pearson correlation with the categorical variable specifying the class of the subject, obtaining a correlation map. Selection of the voxels sites with greatest absolute value of correlation coefficients defines the masks for feature extraction, which are used to build the feature vectors from the individual h -maps. These masks are providing localizations for image biomarkers that may have biomedical significance, therefore we report them separately. Feature vectors are used to perform classification experiments, applying a 10-fold cross validation methodology. We use k-NN classifiers to provide baseline results. Easy to implement and simplicity of k-NN are the important properties that matched our criteria. Furthermore, k-NN can achieve competitive accuracy results even compared to the sophisticated methods as support vector machine, naive Bayes, random forest. Accuracy results are assumed to provide some endorsement of the value of the image biomarkers identified by the feature masks.

2.6 Materials

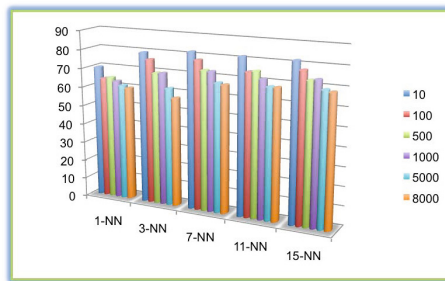
We perform computational experiments resting state fMRI data obtained from a 28 healthy control subjects (NC), and two groups of schizophrenia patients: 26 subjects with and 14 subjects without auditory hallucinations (AH and nAH respectively). For each subject we have 240 BOLD volumes and one T1-weighted anatomical image. Data preprocessing pipeline has been presented in [19,18].



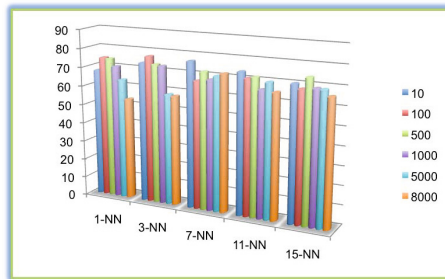
HC vs. Schiz



HC vs. nAH



HC vs. AH



nAH vs. AH

Fig. 2. Maximum Classifier Accuracy found in 10 repetition of 10-fold cross validation for k-NN classifier $k = 1, 3, 7, 11, 15$. The bar colors represent diferent number of extracted features.

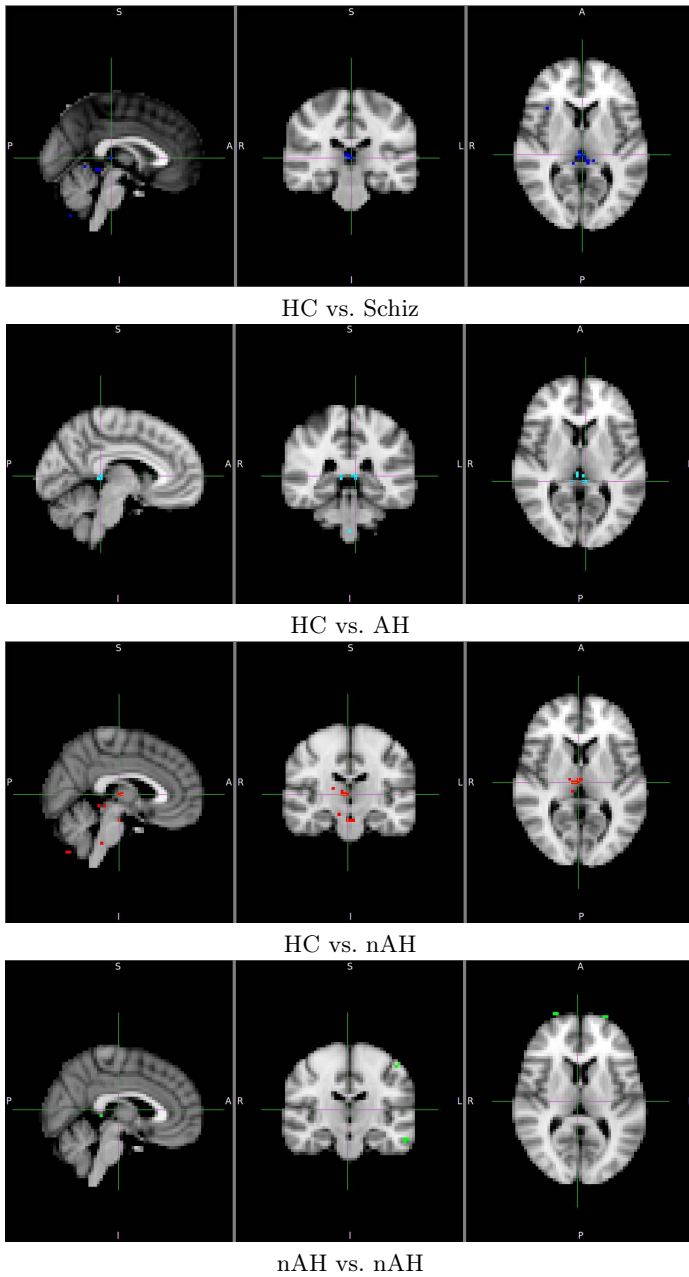


Fig. 3. Visualization of Localization

3 Results

Classification Results

Figure (2) shows the results of the classification experiments on the discrimination the possible pairs of classes: Healthy controls versus Schizophrenia patients (HC vs. Schiz), versus patients without auditory hallucinations (HC vs. nAH), with auditory hallucinations (HC vs. AH), and between classes of patients (nAH vs. AH). The color bars identify the size of the feature vectors, which are built from voxels sites with greatest absolute Pearson's correlation coefficients. In all cases, classification performance decreases with the largest sizes of the feature vectors, which is to be expected because the k-NN classifier suffers from the curse of dimensionality. The best results are obtained in the (HC vs. nAH) case, suggesting that these kind of patients could be better discriminated from healthy controls. Discrimination of the auditory hallucination (nAH vs. AH) is not successful, however we expect that further experimentation will improve results.

Feature Localization in the Brain

Figure (3) shows the voxel sites of the feature extraction in the above enumerated cases. These localizations can be taken as biomarkers for additional research.

4 Conclusions and Future Work

Using the LAAM reconstruction error measured by the Chebyshev distance as a reduced ordering h -map, we define a Foreground/Background/ LAAM-supervised h -map. This data is used on resting state fMRI for the identification of potential biomarkers for schizophrenia and variants with and without auditory hallucinations by Pearson's correlation coefficient with the categorical variable. These biomarkers are evaluated in the terms of the corresponding classification accuracy achieved on the feature vectors extracted from the selected voxel sites. We find that the classification results are encouraging, with best results obtained in the discrimination between healthy controls and patients without auditory hallucinations. Further results will be obtained applying more sophisticated classifier systems to the data. Application of morphological filters to perform feature selection is also considered on the basis of the well defined multivariate mathematical morphology

Acknowledgements. Ann K. Shinn from the McLean Hospital, Belmont, Massachusetts; Harvard Medical School, Boston, Massachusetts, US for providing experimental images. Darya Chyzyk has been supported by a FPU grant from the Spanish MEC. Support from MICINN through project TIN2011-23823. GIC participates at UIF 11/07 of UPV/EHU.

References

1. Craddock, R.C., Holtzheimer, P.E., Hu, X.P., Mayberg, H.S.: Disease state prediction from resting state functional connectivity. *Magnetic Resonance in Medicine* 62(6), 1619–1628 (2009)
2. Northoff, G., Duncan, N.W., Hayes, D.J.: The brain and its resting state activity—experimental and methodological implications. *Progress in Neurobiology* 92(4), 593–600 (2010)
3. van den Heuvel, M.P., Pol, H.E.H.: Exploring the brain network: A review on resting-state fmri functional connectivity. *European Neuropsychopharmacology* 20(8), 519–534 (2010)
4. Cordes, D., Haughton, V., Carew, J.D., Arfanakis, K., Maravilla, K.: Hierarchical clustering to measure connectivity in fmri resting-state data. *Magnetic Resonance Imaging* 20(4), 305–317 (2002)
5. Demirci, O., Stevens, M.C., Andreasen, N.C., Michael, A., Liu, J., White, T., Pearlson, G.D., Clark, V.P., Calhoun, V.D.: Investigation of relationships between fMRI brain networks in the spectral domain using ICA and granger causality reveals distinct differences between schizophrenia patients and healthy controls. *NeuroImage* 46(2), 419–431 (2009)
6. Remes, J.J., Starck, T., Nikkinen, J., Ollila, E., Beckmann, C.F., Tervonen, O., Kiviniemi, V., Silven, O.: Effects of repeatability measures on results of fmri sica: A study on simulated and real resting-state effects. *NeuroImage* (2010) (in press, corrected proof)
7. Calhoun, V.D., Adali, T., Pearlson, G.D., Pekar, J.J.: A method for making group inferences from functional mri data using independent component analysis. *Human Brain Mapping* 14(3), 140–151 (2001)
8. Zou, Q.H., Zhu, C.Z., Yang, Y., Zuo, X.N., Long, X.Y., Cao, Q.J., Wang, Y.F., Zang, Y.F.: An improved approach to detection of amplitude of low-frequency fluctuation (alff) for resting-state fmri: Fractional alff. *Journal of Neuroscience Methods* 172(1), 137–141 (2008)
9. Dosenbach, N.U.F., et al.: Prediction of individual brain maturity using fmri. *Science* 329, 1358–1361 (2010)
10. Pereira, F., Mitchell, T., Botvinick, M.: Machine learning classifiers and fMRI: A tutorial overview. *NeuroImage* 45(1, suppl. 1), S199–S209 (2009); *Mathematics in Brain Imaging*
11. Yao, Z., Wang, L., Lu, Q., Liu, H., Teng, G.: Regional homogeneity in depression and its relationship with separate depressive symptom clusters: A resting-state fmri study. *Journal of Affective Disorders* 115(3), 430–438 (2009)
12. Liu, Y., Wang, K., YU, C., He, Y., Zhou, Y., Liang, M., Wang, L., Jiang, T.: Regional homogeneity, functional connectivity and imaging markers of alzheimer’s disease: A review of resting-state fmri studies. *Neuropsychologia* 46(6), 1648–1656 (2008); *Neuroimaging of Early Alzheimer’s Disease*
13. Mingoia, G., Wagner, G., Langbein, K., Scherpiet, S., Schloesser, R., Gaser, C., Sauer, H., Nenadic, I.: Altered default-mode network activity in schizophrenia: A resting state fmri study. *Schizophrenia Research* 117(2-3), 355–356 (2010); 2nd Biennial Schizophrenia International Research Conference
14. Zhou, Y., Liang, M., Jiang, T., Tian, L., Liu, Y., Liu, Z., Liu, H., Kuang, F.: Functional dysconnectivity of the dorsolateral prefrontal cortex in first-episode schizophrenia using resting-state fmri. *Neuroscience Letters* 417(3), 297–302 (2007)

15. Zhou, Y., Shu, N., Liu, Y., Song, M., Hao, Y., Liu, H., Yu, C., Liu, Z., Jiang, T.: Altered resting-state functional connectivity and anatomical connectivity of hippocampus in schizophrenia. *Schizophrenia Research* 100(1-3), 120–132 (2008)
16. Vercammen, A., Knegtering, H., den Boer, J., Liemburg, E.J., Aleman, A.: Auditory hallucinations in schizophrenia are associated with reduced functional connectivity of the temporo-parietal area. *Biological Psychiatry* 67(10), 912–918 (2010); Anhedonia in Schizophrenia
17. Shinn, A.K., Baker, J.T., Cohen, B.M., Ongur, D.: Functional connectivity of left heschl's gyrus in vulnerability to auditory hallucinations in schizophrenia. *Schizophrenia Research* 143(2-3), 260–268 (2013)
18. Graña, M., Chyzhyk, D.: Hybrid multivariate morphology using lattice auto-associative memories for resting-state fmri network discovery. In: *IEEE 2012 12th International Conference on Hybrid Intelligent Systems, HIS*, pp. 537–542 (2012)
19. Chyzhyk, D., Graña, M.: Results on a lattice computing based group analysis of schizophrenic patients on resting state fMRI. In: Ferrández Vicente, J.M., Álvarez Sánchez, J.R., de la Paz López, F., Toledo Moreo, F. J. (eds.) *IWINAC 2013, Part II. LNCS*, vol. 7931, pp. 131–139. Springer, Heidelberg (2013)
20. Graña, M., Villaverde, I., Maldonado, J., Hernández, C.: Two lattice computing approaches for the unsupervised segmentation of hyperspectral images. *Neurocomputing* 72, 2111–2120 (2009)
21. Grana, M.: A brief review of lattice computing. In: *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008 (IEEE World Congress on Computational Intelligence)*, pp. 1777–1781 (2008)
22. Graña, M.: Lattice computing in hybrid intelligent systems. In: *IEEE Press (ed.) Proc. HIS 2012* (2012)
23. Ritter, G., Sussner, P., Diaz-de-Leon, J.: Morphological associative memories. *IEEE Transactions on Neural Networks* 9(2), 281–293 (1998)
24. Ritter, G., Diaz-de-Leon, J., Sussner, P.: Morphological bidirectional associative memories. *Neural Networks* 12(6), 851–867 (1999)
25. Liu, D., Yan, C., Ren, J., Yao, L., Kiviniemi, V.J., Zang, Y.: Using coherence to measure regional homogeneity of resting-state fmri signal. *Frontiers in Systems Neuroscience* 4(24) (2010)
26. Velasco-Forero, S., Angulo, J.: Supervised ordering in IIR^P: Application to morphological processing of hyperspectral images. *IEEE Transactions on Image Processing* 20(11) 3301–3308 (2011)
27. Ritter, G., Sussner, P., Diaz-de-Leon, J.: Morphological associative memories. *IEEE Transactions on Neural Networks* 9(2), 281–293 (1998)

Enhancing Active Learning Computed Tomography Image Segmentation with Domain Knowledge

Borja Ayerdi, Josu Maiora, and Manuel Graña

Computational Intelligence Group, UPV/EHU

Abstract. This paper follows previous works on the construction of interactive medical image segmentation system, allowing quick volume segmentation requiring minimal intervention of the human operator. This paper contributes to tackle this problem enhancing the previously proposed Active Learning image segmentation system with Domain Knowledge. Active Learning iterates the following process: first, a classifier is trained on the basis of a set of image features extracted for each training labeled voxel; second, a human operator is presented with the most uncertain unlabeled voxels to select some of them for inclusion in the training set assigning corresponding label. Finally, image segmentation is produced by voxel classification of the entire volume with the resulting classifier. The approach has been applied to the segmentation of the thrombus in CTA data of Abdominal Aortic Aneurysm (AAA) patients. The Domain Knowledge referring to the expected shape of the target structures is used to filter out undesired region detections in a post-processing step. We report computational experiments over 6 abdominal CTA datasets consisting. The performance measure is the true positive rate (TPR). Surface rendering provides a 3D visualization of the segmented thrombus. A few Active Learning iterations achieve accurate segmentation in areas where it is difficult to distinguish the anatomical structures due to noise conditions and similarity of gray levels between the thrombus and other structures.

1 Introduction

Active learning. Building a supervised classifier consists in learning a mapping of data features into a set of classes given a labeled training set. Generalization is the ability of providing correct class labels to previously unseen data. Active learning tries to exploit the interaction with a user providing the labels for the training set samples, with the aim of obtaining the most accurate classification using the smallest possible training set. Samples are optimally selected for inclusion, ensuring that they will provide the greatest increase in accuracy [1]. The incremental data selection follows some classification uncertainty criterium that does not require actual knowledge of the data sample label, thus no double dipping is incurred. The human operator provides the labels of the selected data for its inclusion in the training set. Besides economy of computation and data

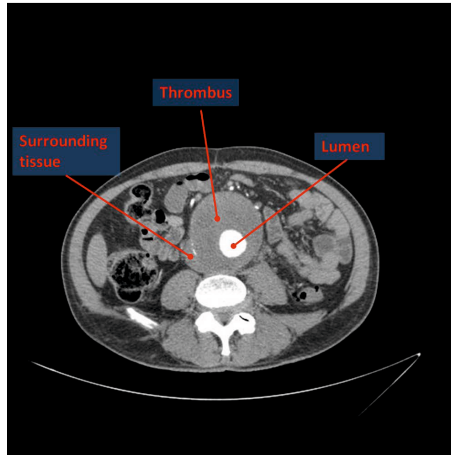


Fig. 1. Axial view of thrombus and lumen in a CTA orthoslice using the contrast agents, blood in lumen is highlighted but thrombus intensity levels are similar to other surrounding tissue

labeling, Active Learning assumes that the underlying data statistics are non stationary, so that the classifier built at one time instant will not be optimal later on.

Image segmentation can be realized as a classification process, each pixel receives a class label according to the associated image features which can be computed from the pixel neighborhood. We perform the pixel classification using random forest (RF) classifier. RF have been applied to delineate the myocardium in 3D ultrasound (US) of adult hearts [2], brain tissue segmentation [3,4], detection of several organs in CT volumes [5,6]. In [7] we provide some first results of the approach proposed in this paper.

Abdominal Aortic Aneurysm (AAA) is a local dilation of the Aorta that occurs between the renal and iliac arteries. The weakening of the aortic wall leads to its deformation and the generation of a thrombus. 3D Contrast Computerized Tomography Angiography (CTA) is the preferred imaging method because it allows minimally invasive visualization of the Aorta's lumen, thrombus and calcifications. segmentation of the AAA thrombus is a specific case of the vascular structure segmentation problem [8] [9,10], which is not trivial due to the low signal intensity contrast between the aneurysm thrombus and its surrounding tissue, as illustrated in Fig. 1. The method by De Bruijne et al.[11] is an interactive contour tracking method for axial slices; Olabarriaga et al. [12] employ a deformable model approach based on a nonparametric statistical grey-level appearance model to determine the deformable model adaptation direction starting from a lumen contour shape interactive segmentation; Zhuge et al. [13] present

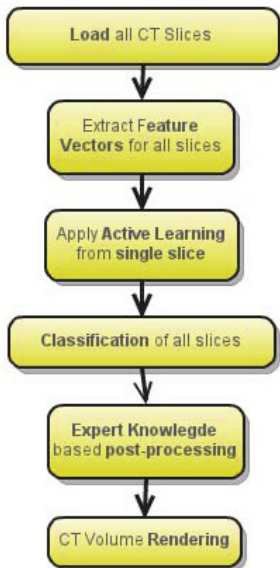


Fig. 2. Pipeline of the experimental setup for the Active Learning enhanced with Domain Knowledge segmentation process

a level-set segmentation based on a parametric statistical model; Demirci et al. [14] propose a deformable B-spline parametric model based on a nonparametric intensity distribution model and; Freiman et al. [15] apply a an iterative model-constrained graph-cut algorithm. These methods involve significant user interaction. Thrombus segmentation of AAA on CTA data volumes can be stated as a voxel classification problem mapping each voxel into either aortic thrombus or background classes.

Process pipeline: The experimental setup is illustrated in Fig. 2. We load the complete CTA volume data, computing first the feature vectors on each voxel. The feature selection and feature extraction processes are the same as in [7,16,17]. A single axial slice situated approximately at the center of the thrombus is selected to perform the Active Learning construction of the voxel classifier. Next, the voxel classifier is applied to all remaining slices of the CTA volume, obtaining an identification of the regions detected as thrombus by this classifier. Expert Domain Knowledge is applied to post-process the detection results, removing spurious detections. Finally, we perform a volume rendering showing the quality of the thrombus detection. The Active Learning oracle in the experiments is the ground truth provided by manual segmentation.

The structure of the paper is as follows: Section 2 describes the methods employed for the segmentation. Section 3 describes the experimental setup. Section 4 provides the experimental results.

2 Methods

2.1 Random Forest Classifiers

The random forests (RF) algorithm is a classifier [18] that encompasses bagging [19] and random decision forests [20] [21], being used in a variety of applications [22]. RF became popular due to its simplicity of training and tuning while offering a similar performance to boosting. Consider a RF collection of tree predictors, that is, a RF is a large collection of decorrelated decision trees

$$h(\mathbf{x}; \psi_t), t = 1, \dots, T,$$

where \mathbf{x} is a d -dimensional random sample of random vector X , ψ_t are independent identically distributed random vectors whose nature depends on their use in the tree construction, and each tree casts a unit vote for the most popular class of input \mathbf{x} . RF capture complex interaction structures in data, and are supposed to be resistant to over-fitting of data if individual trees are sufficiently deep.

Given a dataset of N samples, a bootstrapped training dataset is used to grow tree $h(\mathbf{x}; \psi_t)$ by recursively selecting a random subset of data dimensions \hat{d} such that $\hat{d} \ll d$ and picking the best split of each node based on these variables. Unlike conventional decision trees, pruning is not required. The independent identically distributed random vectors ψ_t determine the random dimension selection.

The trained RF can be used for classification of a new input \mathbf{x} by majority vote among the is the class prediction of the RF trees $C_u(x)$. The critical parameters of the RF classifier for the experiments reported below are set as follows. The number of trees in the forest should be sufficiently large to ensure that each input class receives a number of predictions: we set it to 100. The number of variables randomly sampled at each split node is $\hat{d} = 5$.

2.2 Active Learning

As a first step, image segmentation is produced by a Random Forest (RF) classifier applied on a set of standard image features. The human operator is presented with the most uncertain unlabeled voxels to select some of them for inclusion in the training set, retraining the RF classifier. The approach is applied to the segmentation of the thrombus in CTA data of Abdominal Aortic Aneurysm (AAA) patients. The segmentation is also constrained by knowledge on the expected shape of the target structures. Active learning focuses on the interaction between the user and the classifier. In the context of classifier based image segmentation, the system returns to the user the pixels whose classification outcome is most uncertain. After accurate labeling by the user, pixels are included into the training set in order to retrain the classifier. The classification model is optimized on well-chosen difficult examples, maximizing its generalization capabilities.

2.3 Domain Knowledge

The use of Domain Knowledge allows to post-processing the results of the classification in order to remove spurious detections. This Domain Knowledge consists in the following rules for the specific detection of the thrombus in AAA images:

- At each axial slice, the thrombus is composed of only one connected component. We can remove all connected components disconnected from the one that is more likely to be the thrombus, which is identified by the following rules.
- Thrombus has a roughly circular shape in any axial cut of the volume.
- In successive slices moving away from the thrombus middle slice the radius of the thrombus region decreases.
- In successive axial slices, the thrombus region overlap is large (between 80% and 90% of the area).
- The 2D coordinates of the centroid of the thrombus region have a small (smooth) variation between successive slices.

These rules allow us to perform a heuristic post-processing of the classification results which show a dramatic increase in detection in some cases. These rules do not need any specific parameter tuning and are easily implementable.

3 Experimental Setup

Datasets. We have performed computational experiments over 6 datasets to test the proposed Active Learning enhanced with Expert Knowledge based image classification approach. Each dataset consists in real human contrast-enhanced datasets of the abdominal area with 512x512 pixel resolution on each slice. Each dataset consists of between 216 and 560 slices and 0.887x0.887x1 mm spatial resolution corresponding to patients who suffered Abdominal Aortic Aneurysm. The datasets show diverse sizes and locations of the thrombus. Some of them have metal streaking artifacts due to the stent graft placement. Ground truth segmentations of the thrombus for each dataset that simulates the human oracle providing the labels for the voxels, was obtained manually by a clinical radiologist.

Segmentation problem. We are looking for the segmentation of the thrombus in the AAA formed after the placement of the endo-protesis. Therefore, we deal with a two-class problem.

Parameter tuning. We train the RF classifier with a single slice a to test the sensitivity of the forest parameters: the number of the trees T and their depth D . The increase in performance stabilizes around number of trees = 80 and depth = 20. Once we get the optimal parameters and feature set, we perform the experiment to test our method in the patients CT volumes as illustrated in figure 2.

Validation. The performance measure results of the experiments are the post-processing average True Positive Rate (TPR).

4 Experimental Results

We have performed computational experiments over 6 datasets to test the proposed approach. Each dataset consists in real human contrast-enhanced datasets of the abdominal area with 512x512 pixel resolution on each slice. Each dataset consists of a number of slices between 216 and 560, and 0.887x0.887x1 mm spatial resolution corresponding to patients who suffered Abdominal Aortic Aneurysm. The datasets show diverse sizes and locations of the thrombus. Fig. 3 shows the performance of the Active Learning based image segmentation algorithm for CT volumes of AAA patients, plotting the average True Positive Rate (TPR)

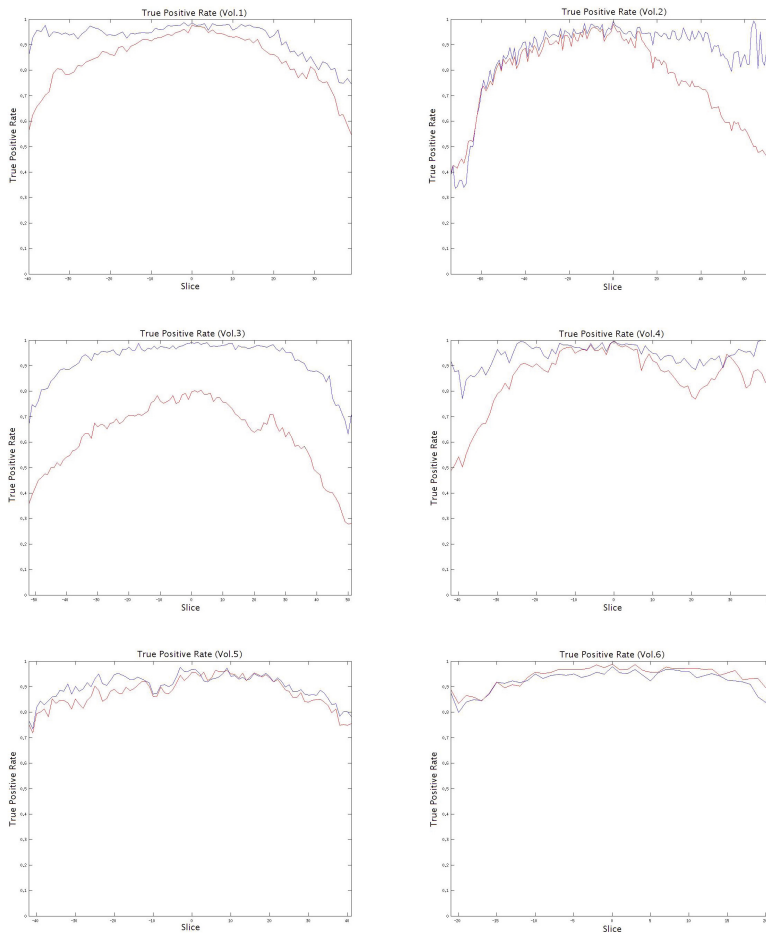


Fig. 3. True Positive Rates for all volumes treated. Red curves corresponds to RF results trained with Active Learning, and blue curves to the Domain Knowledge post-processing.

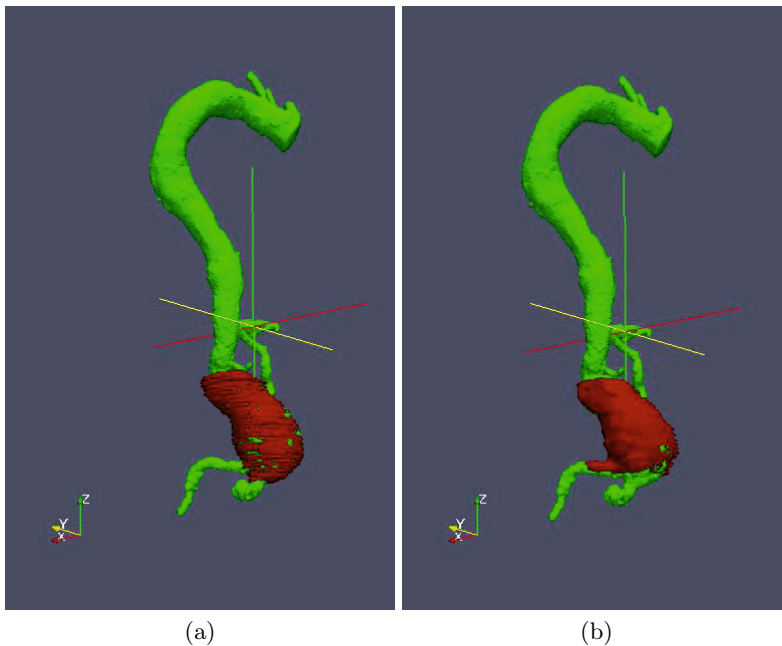


Fig. 4. Volume rendering of aortic lumen (green) and thrombus (red) obtained from the segmentation of one CT volume. (a) manual segmentation of the ground truth, (b) result of Active Learning training of RF classifier, enhanced with Domain Knowledge post-processing rules.

versus the slice number (relative to the middle slice of the thrombus used for training) of the RF classifiers trained with Active Learning without (red) and with (blue) the application of the heuristic postprocessing rules derived from Domain Knowledge. In most of the cases, the Domain Knowledge based post-processing provides some improvement, mostly in the slices that fall far away from the middle slice.

A 3D volume rendering of the Aorta's lumen (green) and thrombus (red) of one patient is shown in Fig.4. Fig.4(a) shows the rendering of the ground truth given by volume manual segmentation. Fig.4 (b) shows the result of the segmentation based on the Active Learning enhanced with Domain Knowledge classifier built from the thrombus' central slice. The structure of the thrombus is well delineated and fits almost perfectly to the ground truth.

5 Conclusion and Future Works

An approach to the problem of segmentation of CTA volumes with a specific application in mind, such as AAA thrombus segmentation treated here, is the training of a voxel based classifier based on selected voxel features. Following conventional procedures, training requires large training data sets which must

be hand-labeled, a costly and error prone process. Besides, there is little guarantee that a classifier trained once will have sustained generalization ability. An alternative is to build quickly and efficiently specific classifiers trained on the volume to be segmented with a minimum number of training samples requiring labeling. Such alternative is provided by Active Learning approaches, which we have applied with some success to AAA thrombus segmentation. However, results can be improved by using specific Domain Knowledge of the structure being segmented. In this paper we have transformed such rules in heuristic post-processing rules. The results show that in some cases, the application of such heuristics can provide dramatic increase in performance, maintaining all the advantages of the Active Learning approach.

Acknowledgements. The CTA images used in the computational experimentes were provided by the group of Leo Joskowicz [15] of the Mount Sinai School of Medicine, New York, NY. Project MICINN grant TIN2011-23823 has supported this work.

References

1. Settles, B.: Active learning literature survey. *Sciences New York* 15(2) (2010)
2. Lempitsky, V., Verhoek, M., Alison Noble, J., Blake, A.: Random forest classification for automatic delineation of myocardium in real-time 3D echocardiography. In: Ayache, N., Delingette, H., Sermesant, M. (eds.) FIMH 2009. LNCS, vol. 5528, pp. 447–456. Springer, Heidelberg (2009)
3. Geremia, E., Menze, B.H., Clatz, O., Konukoglu, E., Criminisi, A., Ayache, N.: Spatial decision forests for MS lesion segmentation in multi-channel MR images. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010, Part I. LNCS, vol. 6361, pp. 111–118. Springer, Heidelberg (2010)
4. Yi, Z., Criminisi, A., Shotton, J., Blake, A.: Discriminative, semantic segmentation of brain tissue in MR images. In: Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (eds.) MICCAI 2009, Part II. LNCS, vol. 5762, pp. 558–565. Springer, Heidelberg (2009)
5. Criminisi, A., Shotton, J., Bucciarelli, S.: Decision forests with long-range spatial context for organ localization in ct volumes. In: MICCAI Workshop on Probabilistic Models for Medical Image Analysis (2009)
6. Criminisi, A., Shotton, J., Robertson, D., Konukoglu, E.: Regression forests for efficient anatomy detection and localization in CT studies. In: Menze, B., Langs, G., Tu, Z., Criminisi, A. (eds.) MICCAI 2010. LNCS, vol. 6533, pp. 106–117. Springer, Heidelberg (2011)
7. Maiora, J., Graña, M.: Abdominal cta image analisis through active learning and decision random forests: Application to AAA segmentation. In: The 2012 International Joint Conference on Neural Networks, IJCNN, pp. 1–7 (2012)
8. Lesage, D., Angelini, E.D., Bloch, I., Funka-Lea, G.: A review of 3d vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical Image Analysis* 13(6), 819–845 (2009)
9. Macia, I., Grana, M., Maiora, J., Paloc, C., de Blas, M.: Detection of type ii endoleaks in abdominal aortic aneurysms after endovascular repair. *Computers in Biology and Medicine* 41(10), 871–880 (2011)

10. Macia, I., Graña, M., Paloc, C.: Knowledge management in image-based analysis of blood vessel structures. *Knowledge and Information Systems* 30(2), 457–491 (2012)
11. de Bruijne, M., van Ginneken, B., Viergever, M.A., Niessen, W.J.: Interactive segmentation of abdominal aortic aneurysms in cta images. *Med. Image Anal.* 8(2), 127–138 (2004)
12. Olabarriga, S., Rouet, J., Fradkin, M., Breeuwer, M., Niessen, W.: Segmentation of thrombus in abdominal aortic aneurysms from CTA with nonparametric statistical grey level appearance modeling. *IEEE Transactions on Medical Imaging* 24(4), 477–485 (2005)
13. Zhuge, F., Rubin, G.D., Sun, S.H., Napel, S.: An abdominal aortic aneurysm segmentation method: Level set with region and statistical information. *Medical Physics* 33(5), 1440–1453 (2006)
14. Demirci, S., Lejeune, G., Navab, N.: Hybrid deformable model for aneurysm segmentation. In: *ISBI 2009*, pp. 33–36 (2009)
15. Freiman, M., Esses, S.J., Joskowicz, L., Sosna, J.: An Iterative Model-Constraint Graph-cut Algorithm for Abdominal Aortic Aneurysm Thrombus Segmentation. In: *Proc. of the 2010 IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro, ISBI 2010, Rotterdam, The Netherlands*, pp. 672–675. IEEE (April 2010)
16. Chyzhyk, D., Ayerdi, B., Maiora, J.: Active learning with bootstrapped dendritic classifier applied to medical image segmentation. *Pattern Recognition Letters* (online, 2013)
17. Maiora, J., Ayerdi, B., Graña, M.: Random forest active learning for computed tomography angiography image segmentation. *Neurocomputing* (inpress, 2013)
18. Breiman, L.: Random forests. *Machine Learning* 45(1), 5–32 (2001)
19. Breiman, L.: Bagging predictors. *Machine Learning* 24(2), 123–140 (1996)
20. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. *Neural Computation* 9(7), 1545–1588 (1997)
21. Ho, T.: The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8), 832–844 (1998)
22. Barandiaran, I., Paloc, C., Graña, M.: Real-time optical markerless tracking for augmented reality applications. *Journal of Real-Time Image Processing* 5, 129–138 (2010)

Evolutionary Ordinal Extreme Learning Machine^{*}

Javier Sánchez-Monedero,
Pedro Antonio Gutiérrez, and Cesar Hervás-Martínez

University of Córdoba, Dept. of Computer Science and Numerical Analysis
Rabanales Campus, Albert Einstein building, 14071 - Córdoba, Spain
{jsanchezm, pagutierrez, chervas}@uco.es

Abstract. Recently the ordinal extreme learning machine (ELMOR) algorithm has been proposed to adapt the extreme learning machine (ELM) algorithm to ordinal regression problems (problems where there is an order arrangement between categories). In addition, the ELM standard model has the drawback of needing many hidden layer nodes in order to achieve suitable performance. For this reason, several alternatives have been proposed, such as the evolutionary extreme learning machine (EELM). In this article we present an evolutionary ELMOR that improves the performance of ELMOR and EELM for ordinal regression. The model is integrated in the differential evolution algorithm of EELM, and it is extended to allow the use of a continuous weighted RMSE fitness function which is proposed to guide the optimization process. This favors classifiers which predict labels as close as possible (in the ordinal scale) to the real one. The experiments include eight datasets, five methods and three specific performance metrics. The results show the performance improvement of this type of neural networks for specific metrics which consider both the magnitude of errors and class imbalance.

Keywords: ordinal classification, ordinal regression, extreme learning machine, differential evolution, class imbalance.

1 Introduction

Ordinal regression, or ordinal classification, problems are classification problems where the problem nature suggests the presence of an order between labels. In addition, it is expected that this order would be reflected on the data distribution through the input space [1]. Compared to nominal classification, ordinal classification has not attracted much attention, nevertheless the number of algorithms and associated publications have grown in the late years [2].

In this work we propose an evolutionary extreme learning machine for ordinal regression. We modify the ELMOR model proposed by Deng et. al [3] with an

^{*} This work has been partially subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P11-TIC-7508 project of the “Junta de Andalucía” (Spain).

extension to allow a probabilistic formulation of the neural network, for which we propose a fitness function that considers restrictions related to ordinal regression problems. We evaluate the proposal with eight datasets, five related methods and three specific performance metrics.

The rest of the paper is organized as follows. Section 2 introduces the ordinal regression problem and formulation. Section 3 presents the extreme learning machine and its evolutionary alternative, and Section 4 explains the proposed method. Experiments are covered at Section 5 and finally conclusions and future work are summarized in the last section.

2 Ordinal Regression

Ordinal regression is a type of supervised classification problem in which there is an order within categories [1,4]. This order is generally deduced from the problem nature by an expert or by simple assumptions about the data.

2.1 Problem Formulation

The ordinal regression problem can be mathematically formulated as a problem of learning a mapping ϕ from an input space \mathbb{X} to a finite set $\mathcal{C} = \{C_1, C_2, \dots, C_Q\}$ containing Q labels, where the label set has an order relation $C_1 \prec C_2 \prec \dots \prec C_Q$ imposed on it (symbol \prec denotes the ordering between different categories). The rank of an ordinal label can be defined as $\mathcal{O}(C_q) = q$. Each pattern is represented by a K -dimensional feature vector $\mathbf{x} \in \mathbb{X} \subseteq \mathbb{R}^K$ and a class label $t \in \mathcal{C}$. The training dataset \mathbf{D} is composed of N patterns $\mathbf{D} = \{(\mathbf{x}_i, t_i) \mid \mathbf{x}_i \in \mathbb{X}, t_i \in \mathcal{C}, i = 1, \dots, N\}$, with $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iK})$.

For instance, bond rating can be considered as an ordinal regression problem where the purpose is to assign the right ordered category to bonds, being the category labels $\{C_1 = \text{AAA}, C_2 = \text{AA}, C_3 = \text{A}, C_4 = \text{BBB}, C_5 = \text{BB}\}$, where labels represent the bond quality assigned by credit rating agencies. Here there is a natural order between classes $\{\text{AAA} \prec \text{AA} \prec \text{A} \prec \text{BBB} \prec \text{BB}\}$, AAA being the highest quality one and BB the worst one.

Considering the previous definitions, an ordinal classifier (and the associated training algorithm) has two challenges. First, since the nature of the problem implies that the class order is somehow related to the distribution of patterns in the space of attributes \mathbb{X} as well as the topological distribution of the classes, the classifier must exploit this a priori knowledge about the input space [1,4]. Secondly, specific performance metrics are needed. Given the bond rating example, it is reasonable to conclude that predicting class BB when the real class is AA represents a more severe error than that associated with AAA prediction. Therefore, performance metrics must consider the order of the classes so that misclassifications between adjacent classes should be considered less important than the ones between non-adjacent classes, more separated in the class order [5,4].

2.2 Performance Metrics

As mentioned, ordinal regression needs specific performance metrics. In this work we will use the accuracy and the Mean Absolute Error (*MAE*), since those are the most used ones, and the recently proposed average *MAE*, which is a robust metric for imbalanced datasets. Let us suppose we want to evaluate the performance of N predicted ordinal labels for a given dataset $\{\hat{t}_1, \hat{t}_2, \dots, \hat{t}_N\}$, with respect to the actual targets $\{t_1, t_2, \dots, t_N\}$. The accuracy, also known as Correct Classification Rate or Mean Zero-One Error (*MZE*) when expressed as an error, is the rate of correctly classified patterns.

However, the *MZE* does not reflect the magnitude of the prediction errors. For this reason, the *MAE* is commonly used together with *MZE* in the ordinal regression literature [2,5,6]. *MAE* is the average absolute deviation of the predicted labels from the true labels:

$$MAE = \frac{1}{N} \sum_{i=1}^N e(\mathbf{x}_i), \quad (1)$$

where $e(\mathbf{x}_i) = |\mathcal{O}(t_i) - \mathcal{O}(\hat{t}_i)|$. The *MAE* values range from 0 to $Q - 1$. However, neither *MZE*, nor *MAE* are suitable for problems with imbalanced classes. To solve this issue, Baccianella et. al [7] proposed to use the average of the *MAE* across classes:

$$AMAE = \frac{1}{Q} \sum_{j=1}^Q MAE_j = \frac{1}{Q} \sum_{j=1}^Q \frac{1}{n_j} \sum_{i=1}^{n_j} e(\mathbf{x}_i), \quad (2)$$

where *AMAE* values range from 0 to $Q - 1$ and n_j is the number of patterns in class j .

3 Extreme Learning Machine

This section presents the ELM and ELMOR models, in order to establish the baseline for the article proposal.

3.1 ELM for Nominal Classification and Regression

This section presents the extreme learning machine (ELM) algorithm and the Evolutionary ELM. For a further review of ELM please refer to specific survey [8]. The ELM algorithm has been proposed in [9]. ELM and its extensions have been applied to several domains including multimedia Quality-of-Service (QoS) [10] or sales forecasting, among others.

The ELM model is a Single-Layer Feedforward Neural Network that is described as follows. Let us define a classification problem with a training set given by N samples $\mathbf{D} = \{(\mathbf{x}_i, \mathbf{y}_i) : \mathbf{x}_i \in \mathbb{R}^K, \mathbf{y}_i \in \mathbb{R}^Q, i = 1, 2, \dots, N\}$, where \mathbf{x}_i is

a $K \times 1$ input vector and \mathbf{y}_i is a $Q \times 1$ target vector¹ Here, a target \mathbf{y} , associated to pattern \mathbf{x} , is defined so that $y_j = 1$ means that pattern \mathbf{x} belong to class j and $y_k = 0|j \neq k$ means the pattern does not belong to class k , this is generally known as a 1-of- Q coding scheme. Let us consider a multi-layer perceptron (MLP) with M nodes in the hidden layer and Q nodes in the output layer given by:

$$f(\mathbf{x}, \boldsymbol{\theta}) = (f_1(\mathbf{x}, \boldsymbol{\theta}_1), f_2(\mathbf{x}, \boldsymbol{\theta}_2), \dots, f_Q(\mathbf{x}, \boldsymbol{\theta}_Q)), \tag{3}$$

where:

$$f_q(\mathbf{x}, \boldsymbol{\theta}_q) = \beta_0^q + \sum_{j=1}^M \beta_j^q \sigma_j(\mathbf{x}, \mathbf{w}_j), q = 1, 2, \dots, Q, \tag{4}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_Q)^T$ is the transpose matrix containing all the neural net weights, $\boldsymbol{\theta}_q = (\boldsymbol{\beta}^q, \mathbf{w}_1, \dots, \mathbf{w}_M)$ is the vector of weights of the q output node, $\boldsymbol{\beta}^q = \beta_0^q, \beta_1^q, \dots, \beta_M^q$ is the vector of weights of the connections between the hidden layer and the q th output node, $\mathbf{w}_j = (w_{1j}, \dots, w_{Kj})$ is the vector of weights of the connections between the input layer and the j th hidden node, Q is the number of classes in the problem, M is the number of sigmoidal units in the hidden layer and $\sigma_j(\mathbf{x}, \mathbf{w}_j)$ the sigmoidal function:

$$\sigma_j(\mathbf{x}, \mathbf{w}_j) = \frac{1}{1 + \exp\left(-\left(w_{0j} + \sum_{i=1}^K w_{ij}x_i\right)\right)}, \tag{5}$$

where w_{0j} is the bias of the j th hidden node.

The linear system $f(\mathbf{x}_j) = \mathbf{y}_j, j = 1, 2, \dots, N$, can be written as the following matrix system $\mathbf{H}\boldsymbol{\beta} = \mathbf{Y}$, where \mathbf{H} is the hidden layer output matrix of the network:

$$H(\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{w}_1, \dots, \mathbf{w}_M) = \begin{bmatrix} \sigma(\mathbf{x}_1, \mathbf{w}_1) & \dots & \sigma(\mathbf{x}_1, \mathbf{w}_M) \\ \vdots & \ddots & \vdots \\ \sigma(\mathbf{x}_N, \mathbf{w}_1) & \dots & \sigma(\mathbf{x}_N, \mathbf{w}_M) \end{bmatrix}_{N \times M},$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_M \end{bmatrix}_{M \times Q} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}_{N \times Q}.$$

The ELM algorithm randomly selects the $\mathbf{w}_j = (w_{1j}, \dots, w_{Kj}), j = 1, \dots, M$, weights and biases for hidden nodes, and analytically determines the output weights $\beta_0^q, \beta_1^q, \dots, \beta_M^q$ for $q = 1 \dots Q$ by finding the least square solution to the given linear system. The minimum norm least-square solution (LS) to the linear system is $\hat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{Y}$, where \mathbf{H}^\dagger is the Moore-Penrose generalized inverse of matrix \mathbf{H} . The minimum norm LS solution is unique and has the smallest norm among all the LS solutions, which guarantees better generalization performance.

¹ Note we change the notation of the targets here from a scalar target (t) to a vector target (\mathbf{y}). This is due to the multi-class neural network outputs, since neural networks generally have Q or $Q - 1$ number of output neurons.

The evolutionary extreme learning machine (EELM) [11] improves the original ELM by using the original Differential Evolution (DE) algorithm proposed by Storn and Price [12]. The EELM uses DE to select the input weights \mathbf{w}_j , and the Moore-Penrose generalized inverse to analytically determine the output weights between hidden and output layers. Then, the population of the evolutionary algorithm is the set of input weights \mathbf{w}_j which are evaluated completing the ELM training process.

3.2 ELM for Ordinal Regression

The ELM has been adapted to ordinal regression by Deng et. al [3] being the key of their approach the output coding strategies that impose the class ordering restriction. That work evaluates single multi-class and multi-model binary classifiers. The single ELM was found to obtain slightly better generalization results for benchmark datasets and also to report the lowest computational time for training. In the present work the single ELM alternative will be used. In the single ELMOR approach the output coding is a targets binary decomposition [13], an example of five classes ($Q = 5$) decomposition is shown in Table 1.

Table 1. Example of nominal and ordinal output coding for five classes ($Q = 5$)

1-of- Q coding	Frank and Hall coding [13]
$\begin{pmatrix} +1 & -1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 & -1 \\ -1 & -1 & +1 & -1 & -1 \\ -1 & -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & -1 & +1 \end{pmatrix}$	$\begin{pmatrix} +1, -1, -1, -1, -1 \\ +1, +1, -1, -1, -1 \\ +1, +1, +1, -1, -1 \\ +1, +1, +1, +1, -1 \\ +1, +1, +1, +1, +1 \end{pmatrix}$

In this way, the solutions provided by the $\hat{\beta} = \mathbf{H}^\dagger \mathbf{Y}$ expression tend to produce order aware models. For the generalization phase, the loss-based decoding approach [14] is applied, i.e. the chosen label is that which minimizes the exponential loss:

$$\hat{t} = \arg \min_{1 \leq q \leq Q} d_L(\mathbf{M}_q, \mathbf{g}(\mathbf{x})),$$

where \hat{t} is the predicted class label, being $\hat{t} \in \mathcal{C} = \{C_1, C_2, \dots, C_Q\}$ containing Q labels, \mathbf{M}_q is the code associated to class C_q (i.e. each of the rows of coding at the right of Table 1), $\mathbf{g}(\mathbf{x}) = f(\mathbf{x}, \boldsymbol{\theta})$ is the vector of predictions given by the model in Eq. (3), and $d_L(\mathbf{M}_q, \mathbf{g}(\mathbf{x}))$ is the exponential loss function:

$$d_L(\mathbf{M}_q, \mathbf{g}(\mathbf{x})) = \sum_{i=1}^Q \exp(-\mathbf{M}_{iq} \cdot g_i(\mathbf{x})). \quad (6)$$

4 Evolutionary Extreme Learning Machine for Ordinal Regression

This section presents our evolutionary extreme learning machine for ordinal regression (EELMOR) model and the associated training algorithm. First, the EELMOR extends the ELMOR model to obtain a probabilistic output. For doing that, the softmax transformation layer is added to the ELMOR model using the negative exponential losses of Eq. (6):

$$p_q = p_q(\mathbf{x}, \boldsymbol{\theta}_q) = \frac{\exp(-d_L(\mathbf{M}_q, \mathbf{g}(\mathbf{x})))}{\sum_{i=1}^Q \exp(-d_L(\mathbf{M}_i, \mathbf{g}(\mathbf{x})))}, \quad 1 \leq q \leq Q, \quad (7)$$

where p_q is the posterior probability that a pattern \mathbf{x} has of belonging to class C_q and this probability should be maximized for the actual class and minimized (or ideally be zero) for the rest of the classes. This formulation is used for evaluating the individuals in the evolutionary process but not for solving the ELMOR system of equations.

In the case of ordinal regression, the posterior probability must decrease from the true class to more distant classes. This has been pointed out in the work of Pinto da Costa et al. [5]. In that work an unimodal output function is imposed to the neural network model, and the probability function monotonically decreases as the classes are more distant from the true one.

According to the previous observation, we propose a fitness function for guiding the evolutionary optimization that simultaneously considers two features of a classifier:

1. Misclassification of non-adjacent classes should be more heavily penalized as the difference between classes labels grows.
2. The posterior probability should be unimodal and monotonically decrease for non-adjacent classes.

In this way, not only the right class output is considered, but also the posterior probabilities with respect to the wrong classes are reduced. In order to satisfy these restrictions, we propose the weighted root mean square error (*WRMSE*).

First, we design the type of cost associated with the errors. Let us define the absolute cost matrix as \mathbf{A} , where the element $a_{ij} = |i - j|$ is equal to the difference in the number of categories, $a_{ij} = |i - j|$. The absolute cost matrix is used, for instance, for calculating the *MAE*, being i the actual label and j the predicted label. An example of an absolute cost matrix for five classes is shown in Table 2. In the case of *WRMSE*, \mathbf{A} cannot be directly applied because it would suppress information about the posterior probability of the correct class (see Eq. (9)). Then, we add a square matrix of ones $\mathbf{1}$ so that our final cost matrix is $\mathbf{C} = \mathbf{A} + \mathbf{1}$ (see an example in Table 2).

Second, according to the model output defined in Eq. (7), we define the weighted root mean square error (*WRMSE*) associated to a pattern as:

$$e = \frac{\sum_{q=1}^Q (c_{iq} \sqrt{(y_q - p_q)^2})}{Q}, \quad (8)$$

Table 2. Example of an absolute cost matrix (\mathbf{A}) and an absolute cost matrix plus the matrix of ones ($\mathbf{C} = \mathbf{A} + \mathbf{1}$) for five classes ($Q = 5$)

\mathbf{A}	$\mathbf{C} = \mathbf{A} + \mathbf{1}$
$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 2 & 3 \\ 4 & 3 & 2 & 1 & 2 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix}$

where i is index of the true target and c_{iq} represents the cost of errors associated to the q output of the neural network coded in matrix \mathbf{C} (see Table 2). Finally, the total error of the prediction is defined as:

$$WRMSE = \frac{\sum_{i=1}^N (e_i)}{N}. \quad (9)$$

For ending this section, it should be noticed that in a single model multi-class classifier the $RMSE$ has the interesting property of selecting solutions that consider good classification performance of all classes simultaneously [15]. In the case of MZE , only one network output (the one with maximum value) contributes to the error function, and it does not contribute with the output's value. However, for $RMSE$ it is straightforward to check that each model output (posterior probabilities) contributes to the error function. Then, the model's decision thresholds and posteriors will tend to be more discriminative. This implicit pressure over the posteriors is even more severe in the case of $WRMSE$.

5 Experimental Section

This section presents experiments comparing the present approach with several alternatives, with special attention to the EELM and the ELMOR as reference methods.

5.1 Datasets and Related Methods

Table 3 shows the characteristics of the 8 datasets included in the experiments. The publicly available real ordinal regression datasets were extracted from benchmark repositories (UCI [16] and `mldata.org` [17]). The experimental design includes 30 stratified random splits (with 75% of patterns for training and the remainder for generalization).

In addition to the EELM, ELMOR and the proposed method (EELMOR), we include the following alternatives in the experimental section:

- The POM algorithm [18], with the *logit* link function.
- The GPOR method [6] including automatic relevance determination, as proposed by the authors.

Table 3. Characteristics of the benchmark datasets

Dataset	#Pat.	#Attr.	#Classes	Class distribution
automobile (AU)	205	71	6	(3, 22, 67, 54, 32, 27)
balance-scale (BS)	625	4	3	(288, 49, 288)
bondrate (BO)	57	37	5	(6, 33, 12, 5, 1)
contact-lenses (CL)	24	6	3	(15, 5, 4)
eucalyptus (EU)	736	91	5	(180, 107, 130, 214, 105)
LEV (LE)	1000	4	5	(93, 280, 403, 197, 27)
newthyroid (NT)	215	5	3	(30, 150, 35)
pasture (PA)	36	25	3	(12, 12, 12)

- NNOR [19] Neural Network with decomposition scheme by Frank and Hall in [13].

The algorithms' hyper-parameters were adjusted by a grid search using *MAE* as parameter selection criteria. For NNOR, the number of hidden neurons, M , was selected by considering the following values, $M \in \{5, 10, 20, 30, 40\}$. The sigmoidal activation function was considered for the hidden neurons. For ELMOR, EELM and EELMOR, higher numbers of hidden neurons are considered, $M \in \{5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, given that it relies on sufficiently informative random projections [9]. With regards to the GPOR algorithm, the hyperparameters are determined by part of the optimization process. For EELM and EELMOR the evolutionary parameters' values are the same as used at [11]. The number of iterations was 50 and the population size 40.

5.2 Experimental Results

Table 4 shows mean generalization performance of all the algorithms including metrics described at Section 2.2. The mean rankings of *MZE*, *MAE* and *AMAE* are obtained to compare the different methods. A Friedman's non-parametric test for a significance level of $\alpha = 0.05$ has been carried out to determine the statistical significance of the differences in rank in each method. The test rejected the null-hypothesis stating that all algorithms performed equally in the mean ranking of the three metrics. Because of space restrictions, we will only examine *AMAE* metric, since it is the most robust one. For this purpose, we have applied the Holm post-hoc test to compare EELMOR to all the other classifiers in order to justify our proposal. The Holm test is a multiple comparison procedure that works with a control algorithm (EELMOR) and compares it to the remaining methods [20]. Results of the test are shown in Table 5, which shows that our proposal improves on all the methods' performance except NNOR for $\alpha = 0.10$, and there are only statistical differences with EELM for $\alpha = 0.05$. The second best performance in *AMAE* was for NNOR.

Table 4. Experimental generalization results comparing the proposed method to other nominal and ordinal classification methods. The mean and standard deviation of the results are reported for each dataset, as well as the mean ranking. The best result is in bold face and the second best result in italics.

Method/DataSet	MZE Mean								Mean MZE rank
	AU	BS	BO	CL	EU	LE	NT	PA	
EELM	0.453	0.152	0.544	0.344	0.507	0.393	0.152	0.389	4.94
ELMOR	0.384	0.082	<i>0.476</i>	0.383	0.440	0.371	0.051	0.389	3.31
GPOR	0.389	0.034	0.422	0.394	0.315	0.388	<i>0.034</i>	0.478	3.13
NNOR	<i>0.376</i>	<i>0.039</i>	0.500	0.294	0.418	0.373	0.035	0.237	2.31
POM	0.533	0.092	0.656	0.378	0.841	0.380	0.028	0.504	4.69
EELMOR	0.360	0.092	0.533	<i>0.306</i>	<i>0.394</i>	<i>0.372</i>	0.035	<i>0.333</i>	<i>2.63</i>
Method/DataSet	MAE Mean								Mean MAE rank
	AU	BS	BO	CL	EU	LE	NT	PA	
EELM	0.688	0.216	0.722	0.517	0.718	0.439	0.154	0.404	5.06
ELMOR	0.542	0.089	0.649	0.522	0.531	0.406	0.052	0.404	3.44
GPOR	0.594	0.034	0.624	0.511	0.331	0.422	<i>0.034</i>	0.489	2.75
NNOR	0.503	<i>0.044</i>	0.671	<i>0.456</i>	0.476	0.408	0.035	0.241	<i>2.44</i>
POM	0.953	0.111	0.947	0.533	2.029	0.415	0.028	0.585	5.00
EELMOR	<i>0.510</i>	0.108	<i>0.644</i>	0.433	<i>0.447</i>	<i>0.407</i>	0.035	<i>0.344</i>	2.31
Method/DataSet	AMAE Mean								Mean AMAE rank
	AU	BS	BO	CL	EU	LE	NT	PA	
EELM	0.813	0.426	1.119	0.545	0.778	0.632	0.212	0.404	4.75
ELMOR	0.649	0.176	1.168	0.531	0.575	0.611	0.114	0.404	3.94
GPOR	0.792	0.051	1.360	0.651	0.362	0.654	0.062	0.489	4.13
NNOR	0.566	<i>0.066</i>	1.135	<i>0.493</i>	0.506	0.608	0.059	0.241	<i>2.19</i>
POM	1.026	0.107	<i>1.103</i>	0.535	1.990	0.632	0.050	0.585	4.06
EELMOR	<i>0.592</i>	0.172	1.041	0.463	<i>0.489</i>	0.608	<i>0.052</i>	<i>0.344</i>	1.94

Table 5. Table with the different algorithms compared with EELMOR using the Holm procedure ($\alpha = 0.10$) in terms of *AMAE*. The horizontal line shows the division between methods significantly different from EELMOR.

i	Algorithm	z	p	α_{Holm}
1	EELM	3.0067	0.0026	0.0200
2	GPOR	2.3385	0.0194	0.0250
3	POM	2.2717	0.0231	0.0333
4	ELMOR	2.1381	0.0325	0.0500
5	NNOR	0.2673	0.7893	0.1000

6 Conclusions and Future Work

In this work, we have adapted the ELMOR model to the Evolutionary ELM. We have proposed the weighed RMSE error function to guide the algorithm. Based on theoretical analysis and experimental results, we justify the proposal compared to the reference methods and other ordinal regression techniques.

Future work involves the design and experiments with new output codes and associated error functions. In addition, as a future work, a comparison can be performed taking into account the run time of the algorithms. Also the exploration of limitations of the proposal should be part of future research.

References

1. Hühn, J.C., Hüllermeier, E.: Is an ordinal class structure useful in classifier learning? *Int. J. of Data Mining, Modelling and Management* 1(1), 45–67 (2008)
2. Gutiérrez, P.A., Pérez-Ortiz, M., Fernández-Navarro, F., Sánchez-Monedero, J., Hervás-Martínez, C.: An Experimental Study of Different Ordinal Regression Methods and Measures. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) *HAIS 2012, Part II. LNCS*, vol. 7209, pp. 296–307. Springer, Heidelberg (2012)
3. Deng, W.Y., Zheng, Q.H., Lian, S., Chen, L., Wang, X.: Ordinal extreme learning machine. *Neurocomputing* 74(1-3), 447–456 (2010)
4. Sánchez-Monedero, J., Gutiérrez, P.A., Tiño, P., Hervás-Martínez, C.: Exploitation of Pairwise Class Distances for Ordinal Classification. *Neural Computation* 25(9), 2450–2485 (2013)
5. Pinto da Costa, J.F., Alonso, H., Cardoso, J.S.: The unimodal model for the classification of ordinal data. *Neural Networks* 21, 78–91 (2008)
6. Chu, W., Ghahramani, Z.: Gaussian processes for ordinal regression. *Journal of Machine Learning Research* 6, 1019–1041 (2005)
7. Baccianella, S., Esuli, A., Sebastiani, F.: Evaluation measures for ordinal regression. In: *Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications, ISDA 2009, San Mateo, CA*, pp. 283–287 (2009)
8. Huang, G.B., Wang, D., Lan, Y.: Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics* 2(2), 107–122 (2011)
9. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 42(2), 513–529 (2012)
10. Chen, L., Zhou, L., Pung, H.: Universal Approximation and QoS Violation Application of Extreme Learning Machine. *Neural Processing Letters* 28, 81–95 (2008)
11. Zhu, Q.Y., Qin, A., Suganthan, P., Huang, G.B.: Evolutionary extreme learning machine. *Pattern Recognition* 38(10), 1759–1763 (2005)
12. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4), 341–359 (1997)
13. Frank, E., Hall, M.: A simple approach to ordinal classification. In: Flach, P.A., De Raedt, L. (eds.) *ECML 2001. LNCS (LNAI)*, vol. 2167, pp. 145–156. Springer, Heidelberg (2001)
14. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. *J. of Machine Learning Research* 1, 113–141 (2001)
15. Sánchez-Monedero, J., Gutiérrez, P.A., Fernández-Navarro, F., Hervás-Martínez, C.: Weighting efficient accuracy and minimum sensitivity for evolving multi-class classifiers. *Neural Processing Letters* 34(2), 101–116 (2011)
16. Asuncion, A., Newman, D.: *UCI machine learning repository* (2007)
17. PASCAL: Pascal (Pattern Analysis, Statistical Modelling and Computational Learning) machine learning benchmarks repository (2011), <http://mldata.org/>
18. McCullagh, P., Nelder, J.A.: *Generalized Linear Models*, 2nd edn. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC (1989)
19. Cheng, J., Wang, Z., Pollastri, G.: A neural network approach to ordinal regression. In: *Proceedings of the IEEE International Joint Conference on Neural Networks, IJCNN 2008*, pp. 1279–1284. IEEE Press (2008)
20. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* 7, 1–30 (2006)

Arm Orthosis/Prosthesis Control Based on Surface EMG Signal Extraction

Aaron Suberbiola, Ekaitz Zulueta, Jose Manuel Lopez-Guede,
Ismael Etxeberria-Agiriano, and Bren Van Caesbroeck

University of the Basque Country (UPV/EHU), Vitoria, Spain
{asuberviola001, ekaitz.zulueta, jm.lopez,
ismael.etxeberrria}@ehu.es

Abstract. The goal of this paper is to show EMG based system control applied to motorized orthoses. Through two biometrical sensors it captures biceps and triceps EMG signals, which are then filtered and processed by an acquisition system. Finally an output/control signal is produced and sent to the actuators, which will then perform the proper movement. The research goal is to predict the movement of the lower arm through the analysis of EMG signals, so that the movement can be reproduced by an arm orthosis, powered by two linear actuators.

Keywords: Orthosis, Prosthesis, Control, EMG, Power assistance.

1 Introduction

Due to different reasons, many people have disabilities in their body, and this causes difficulties in their life. Medical science has worked a lot on trying to solve these drawbacks, but sometimes it is impossible to achieve more improvements just with medical treatments. On this paper we have tried to develop an arm orthosis control, using EMG signals as input and creating a movement, as natural as possible, for a robotic arm.

At the University of Tsukuba the Hybrid Assisted Limb (HAL) was developed [1-3]. It is a battery-powered suit that detects muscle myoelectrical signals on the skin surface, below the hip and above the knee. Using these and other signals, such as gyroscopes, force sensors and potentiometers for measuring joint angles, it processes everything and each leg of HAL is powered in flexion/extension motion. The ankle includes passive degrees of freedom.

Yamamoto et al. [4, 5] have created an exoskeleton system for assisting nurses during patient handling. It includes pneumatic actuators for the flexion/extension of the hips and knees. User input is determined via force sensing resistors coupled to the wearer's skin, and the data used comes from those force sensing resistor and joint angles.

Pratt et al. developed a squatting assisting system that powered the knee movement [6]. The device is powered by a linear series-elastic actuator, and it uses a positive-feedback force controller to create an appropriate force for the actuator.

Kong et al. developed a full lower-limb exoskeleton system that works with a powered walker [7]. The exoskeleton is lighter than others, because the electric actuators, the controller and the batteries are placed in the walker. The system's input is a set of pressure sensor that measure the force applied by the quadriceps on the knee.

Agrawal et al. have researched projects on statically balanced leg orthoses that reduce the effort during swing [8]. The device uses springs in order to cancel the gravity force associated with the device links and the person's leg. A substantial reduction of the required torque has been proved experimentally.

Just in the USA there is an estimate of 10,000 new upper extremity amputees every year. This can be caused by trauma, disease or due to congenital deficiencies. When a person loses control over a lower limb, there are several options that can be explored, taking into account several factors such as price, weight and performance. Passive prosthesis can be found among the most popular options. The most basic form, controlled by another limb, has limited possibilities but it is well designed esthetically and has low cost. Similarly, in mechanical ones the movement is controlled by another muscle and transferred by strings, with low cost. Voice controlled alternatives are not common, and have acceptable cost, but have limited control and background noise. EMG signals (electromyography), electrical power generated by remaining muscles of the harmed arm, have high cost and learning curve. Finally, options using AMG/MMG signals (acoustic/mechanical myographic), use the sound produced by contracting muscles, they cost less than EMG, they are not affected by electrical interference, background noise and difficulty of recording only the muscle sound.

2 EMG Signal Acquisition

The electromyography signals (EMG) detect the electrical potential generated by muscle fibers. When muscles are relaxes they generate no potential and when they are flexed to the maximum, the electrical potential takes also the peak value.

It is also important to notice that EMG signals are a combination of several Motor Unit Action Potentials (MUAP), as muscle fibers behave as motor units. The combination of those MUAPs is called Compound Muscle Action Potential (CMAP). Multiple MUAPs can be detected using one single electrode, and the EMG signal must be decomposed using some advanced techniques.

Typical EMG signal values are $50\mu\text{V}$ - 30mV electrical potential and 7-20Hz frequency.

2.1 EMG Signals Capturing

Most common methods of capturing EMG signals are using surface, needle or fine-wire electrodes. Surface electrodes detect a larger number of motor units, while the two other methods allow focusing on single muscle fibers. Additionally, the correct placement of the electrodes affects the results of the EMG measuring.

Comparing surface electrodes, dry and wet ones can be differentiated. Needle electrodes are typically used in physiotherapy, and fine-wire electrodes can be surgically implanted into the muscle. In addition, when using surface electrodes, they can be

mainly placed longitudinally, following the long axis of the muscle, or transversally, perpendicular to the long axis.

2.2 EMG Signals Preparation

Once the EMG electrodes are placed, there are a few features to consider. First of all, voltage potential varies for each individual, so data normalization is very useful [9].

Principal sources of noise that must be avoided are: equipment noise (the higher the quality of the equipment is, the less the noise it generates), ambient noise (electromagnetic radiation caused by electronic devices around) and motion artifact (movement of the electrode cable or the electrode itself may produce irregularities in the data [10]).

Besides, some factors affect the EMG signals. One of the most important is causative factors, which can be extrinsic, like the structure or the placement of the electrodes, or intrinsic, like physiological or anatomical issues. There are also intermediate factors, such as physical and physiological phenomena influenced by causative factors. Finally some deterministic factors must be mentioned, because the number of active motor units and mechanical interaction between muscles are important too.

Extra attention is required by the crosstalk. It has to be taken into account very carefully because it affects the signals that will be processed later. Despite that EMG signals are dominated by the closest muscle, neighbor muscle signals may crosstalk with the desired muscle signals [11].

The effect of crosstalk can be minimized by choosing appropriate size of the electrode conductive area and appropriate inter-electrode distance. Crosstalk may also be further reduced by a proper location of the surface electrodes on the muscle [12]. Care should be taken to place the electrodes on the center of the muscle, away from the borders, although this is not always possible.

3 EMG Signal Processing

Once an EMG signal has been properly prepared and recorded, processing is required to extract as much data as possible. It is important to choose the best feature selection method before starting the signal classification.

3.1 Feature Selection

Depending on the type of data and its origin, different commonly used analysis techniques exist [13], as shown in Table 1.

The Principal Component Analysis (PCA) is interesting for pattern recognition because it reduces the number of coefficients needed for an effective feature representation by discarding the terms with small variances. Factor analysis is used to study the patterns of relationship among dependent variables, so you can discover something about the independent variable that affects them. The only variances that are analyzed are the ones that share variances, so the underlying structure of the variables can be identified [14-16].

Table 1. Selection chart for analysis

	Density estimation / model $P(x)$ / probabilistic model	Define a subspace directly / reduce data / not probabilistic
Data assumed in a subspace	Factor analysis	PCA
Data assumed in groups	Mixture of Gaussians	K-means

K-means clustering assigns a set of samples into a subset called “cluster”, in such a way that samples in the same cluster are alike in some way. It is considered a form of unsupervised learning and it is used in several fields among which can be found pattern recognition [17]. Finally, the mixture of Gaussians assumes that the data is produced by a mixture of N multivariate Gaussians. As Gaussians are to probability densities what sines and cosines are to periodic signals, in theory any distribution can be described as a combination of Gaussians [18-20].

It is also important to understand how the auto regressive model works. It is often used to predict and model various types of phenomena, and due to the stochastic nature of the EMG signals, it is a good solution for estimating signal samples as linear combinations of previous samples. But the reason why auto regressive model is important in our work is not the estimating aspect, but the vector with the AR-parameters that characterizes the data. And this vector is used as input for the classifier.

3.2 Signal Classification

Once the features of the EMG signals have been extracted and selected, the signals themselves have to be classified. The most popular methods for classification are based on artificial intelligence methods such as neural networks, fuzzy networks or neural-fuzzy networks.

Artificial neural networks (ANN) are, in essence, a simulation of how our brain works. They are structures of parallel processing based on the biological brain processing model [21]. It is formed by simple computation elements, which are partial or totally interconnected. Basically it is a network of nodes, in which each node is connected to each other by links. Additionally, each node has a weight value assigned. Artificial neural networks can have one or multiple layers: the more layers they have, the more complicated the problem they solve can be.

4 Practical Experiment

4.1 Setup

The surface electrode used is the Biometrics SX230. It contains all the necessary gain and filters so the biceps and triceps signals can be captured. Electrodes are connected to the Biometrics K800 base unit, and it also has a ground reference cable (R206) so the system has a good reference.

The K800 amplifier system being used has two main parts: the larger table mounted base unit and the small light weight subject unit. The sensors are connected to the subject unit, which has 8 instrumental amplifiers, and it converts all inputs to digital signals and samples the data. Then the data is transferred to the base unit, which converts the signals back to analog for output to proprietary A/D systems.

The data acquisition board used is an AD622. It is used to connect PC compatible computers to real world signals. A PC with a numerical computing environment programming language is also needed.

In this experiment, the arm orthosis shown in Figure 1 is used. It is powered by two linear actuators which control the extension and flexion of the elbow. The actuators used are Firgelli Miniature Linear Motion Series L12.

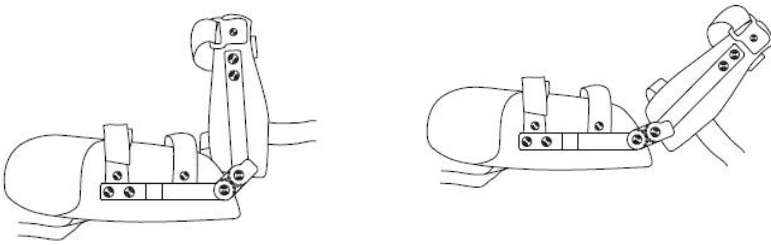


Fig. 1. Custom made arm orthosis

In Figure 2 we can see a general scheme of the tool used during the experiment.



Fig. 2. Practical setup

4.2 Experiment Protocol

In the explained experiment, samples from two healthy male subjects (aged 23 and 25) are collected. The first electrode was positioned on the belly of the biceps brachii, and the second electrode is placed on the triceps brachii opposed to the first electrode. Both electrodes are positioned longitudinally.

The movements that are recorded in this experiment are 10 elbow extensions and 10 flexions. The movements are performed under two different situations: while standing and holding an object of 1kg to apply a minimum level of force to ensure the registration of muscle activity during the movements (dataset A), and seated applying different levels of force varying between minimum and maximum voluntary contraction (dataset B).

The movement speed was varied due to 4 normal movement ranges, 3 fast and 3 slow. Each movement was sampled with a sample time of 1ms.

4.3 Binary Algorithm

The most basic solution is the binary algorithm, which only takes into account the amplitude of the signal. Once the biceps or triceps signal crosses a certain threshold, the orthosis will open or close.

After the signal is recorded, an average filter is applied, and then the signal is rectified and filtered again to accent peaks even more. Finally an envelope detector is used to smooth the final signal. This process is shown in Figure 3.

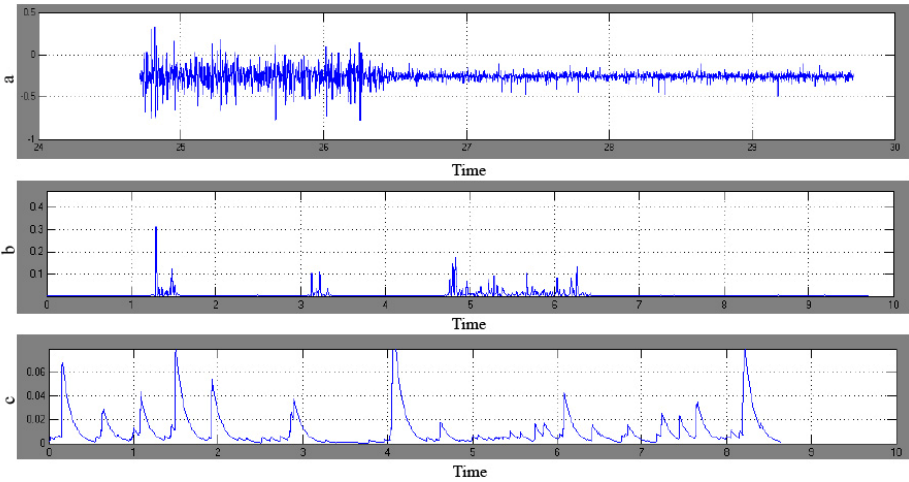


Fig. 3. a) EMG signal b) result of average filter c) result of envelope detector

As it is shown in Figure 4, depending on which threshold values the EMG signal crosses, the orthosis moves up or down.

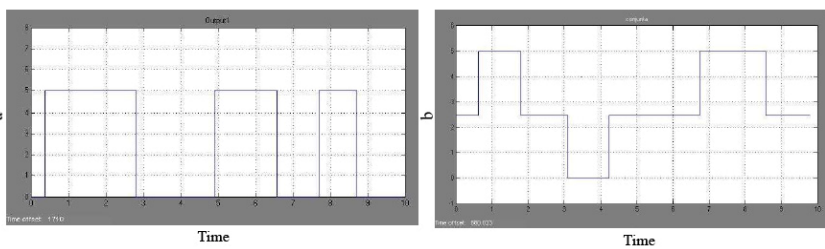


Fig. 4. a) One channel output b) Two channel output

4.4 Variable Algorithm

The variable algorithm uses the same filtering step as the binary algorithm, but it adds a series of threshold values to determine the speed of the movement. At the end the signal is sent through a first order transfer function to smooth the results. The values used in this variable algorithm are shown in Table 2.

Table 2. Variable speed algorithm

Input (X)	Output (Y)	Input (X)	Output (Y)
$40 \geq X < 50$	Y = 100%	$15 \geq X < 20$	Y = 50%
$35 \geq X < 40$	Y = 90%	$10 \geq X < 15$	Y = 40%
$30 \geq X < 35$	Y = 80%	$7.5 \geq X < 10$	Y = 30%
$25 \geq X < 30$	Y = 70%	$5 \geq X < 7.5$	Y = 20%
$20 \geq X < 25$	Y = 60%	$1.5 \geq X < 5$	Y = 10%

However, because of the high amount of force required to cross certain threshold amplitude, the system is difficult to control. In addition, the system needs practice and learning to control it properly. A typical output using the variable algorithm is shown in Figure 5.

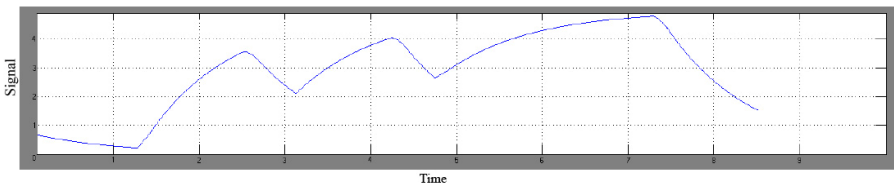


Fig. 5. One channel output

4.5 Autoregressive and Neural Networks

Another algorithm, a bit more complicated than the previous, is the one which uses autoregressive model and neural networks. In Figure 6 the performance of the neural network classifier is shown.

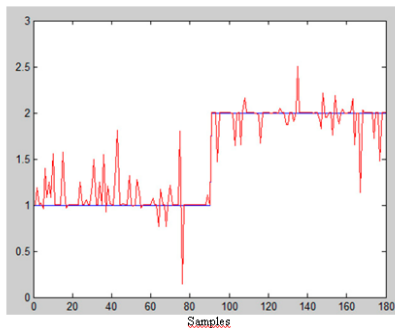


Fig. 6. Performance of the neural network classifier

First of all, after recording the data, the autoregressive coefficients are calculated. A length of 100ms is established for this experiment. Then the neural network is trained several times until the error rate stays below the established value. For the training phase a back propagation network with 15 neurons in the hidden layer is used, with a weight based learning function.

In the last step, the error of the system is calculated. The best performance achieved was a 94.34% accuracy using a sample block length of 100ms and an AR-model of order 15. These values returned from the dataset A.

When the values of dataset B were used, different results were returned. The best performance came from an AR-model of order 4 and block length of 100ms, getting an 80% in the best cases. In this case, lower AR-model order returns a better accuracy level. The results of this experiment are shown in Figure 7.

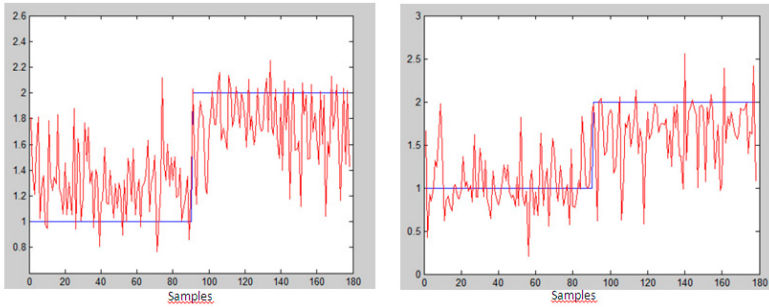


Fig. 7. Neural network response. a) AR order 10. b) AR order 4.

On this highly variance dataset, best results were obtained using an order 4 AR model and a 100ms block length. The network performed around 80% in best cases, being a 5%, on average, better the order 4 AR model rather than the order 10 AR model.

However, it was noticed that the signals toward the end of a movement performed better than the signals created at the beginning of the movement. The performance increased an 11%, getting a 91% of accuracy. In this case, the behavior of the system when the AR-model order is changed improves with higher order. AR-model of order 10 gives 5% more than AR-model of order 4. These results are shown in Figure 8.

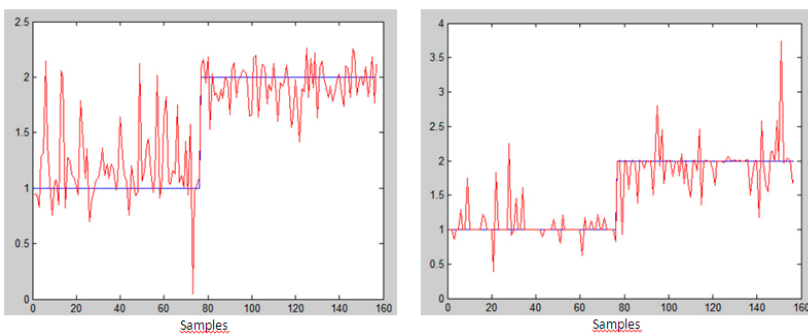


Fig. 8. Neural network response. a) AR order 10. b) AR order 4.

This change in performance can be explained by the fact that triceps signal is much clearer at the end of an elbow extension, and it is only in the later stages of the movement that triceps muscle gives a clear signal. Therefore, we can conclude that

using a highly detailed AR-model only confuses the AR-algorithm when the signals have great variations within the same set. A highly detailed model should only be used when there is a high quality dataset.

5 Conclusions

The binary and variable algorithms are very basic time domain analysis solutions, with a low precision and they require the user to apply a significant amount of force, in order to have a clear detection. Furthermore, the system has to be calibrated for each individual. Despite these disadvantages, the system is very simple, which makes these methods easy to implement, requiring less computational power and a minimum amount of hardware.

The use of the autoregressive model combined with the artificial neural network allows a more precise detection of movement with a minimum amount of force to be applied by the user. It classifies the movement as an elbow flexion or extension, and gives back more accurate results than the previous algorithms.

In order to be suitable as a real life application, it is not only important to compare the accuracy of the classification, but also the response time and complexity of the system. It is also important to keep the system user friendly, in such a way that the training and calibration of the system should be kept to a minimum.

Acknowledgments. The authors belong to Computational Intelligence Group of the University of the Basque Country (UPV/EHU), supported by the Basque Government.

References

1. Kawamoto, H., Sankai, Y.: Power assist system HAL-3 for gait disorder person. In: Miesenberger, K., Klaus, J., Zagler, W.L. (eds.) ICCHP 2002. LNCS, vol. 2398, pp. 196–203. Springer, Heidelberg (2002)
2. Kawamoto, H., Kanbe, S., Sankai, Y.: Power assist method for HAL-3 estimating operator's intention based on motion information. In: IEEE Workshop on Robot and Human Interactive Communication (Millbrae), pp. 67–72 (2003)
3. Kawamoto, H., Suwoong, L., Kanbe, S., Sankai, Y.: Power assist method for HAL-3 using EMG-based feedback controller. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 1648–1653 (2003)
4. Yamamoto, K., Hyodo, K., Ishii, M., Matsuo, T.: Development of power assisting suit for assisting nurse labor. JSME International Journal Series, 703–711 (2002)
5. Yamamoto, K., Hyodo, K., Ishii, M., Yoshimitsu, T., Matsuo, T.: Development of power assisting suit. JSME International Journal Series, 923–930 (2003)
6. Pratt, J.E., Krupp, B.T., Morse, C.J., Collins, S.H.: The RoboKnee: An exoskeleton for Enhancing Strength and Endurance During Walking. In: IEEE International Conference on Robotics and Automation (New Orleans), pp. 2430–2435 (2004)
7. Kong, K., Jeon, D.: Design and control of an exoskeleton for the elderly and patients. IEEE/ASME Transactions on Mechatronics, 220–226 (2006)

8. Agrawal, S.K., Fattah, A.: Theory and design of an orthotic device for full or partial gravity-balancing of a human leg during motion. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 157–165 (2004)
9. Day, S.: Important factors in surface EMG measurement. *Bortec* (2009)
10. Reaz, M.B.I., Hussain, M.S., Mohd-Yasin, F.: Techniques of EMG analysis: detection, processing, classification and applications. *Biological Procedures* (2006)
11. Hug, F.: Can muscle coordination be precisely studied by surface electromyography? *Journal of Electromyography and Kinesiology* 21, 1–12 (2011)
12. Hermens, H.J., Freriks, B., Disselhorst-Klug, C., Rau, G.: Development of recommendations for SEMG sensor placement procedures. *Journal of Electromyography and Kinesiology* 10, 367–374 (2000)
13. Ng, A.Y.: Lecture on machine learning: principal component analysis and independent component analysis in relation to unsupervised machine learning, Stanford (2008)
14. Havran, C., Hupet, L., Czyz, J., Lee, J., Vandendorpe, L., Verleysem, M.: Independent component analysis for face authentication. In: *Knowledge-based Intelligent Information Engineering Systems & Allied Technologies*. IOS Press, Crema (2009)
15. Agrawal, A.: Independent component analysis vs factor analysis. *ENEE698A Seminar* (2003)
16. Ripley, B.: *Principal component analysis and factor analysis*. University of Oxford: Department of Statics (2009)
17. Hill, T., Lewicki, P.: *Statistics: methods and applications. A comprehensive reference for science, industry and data mining*. Statsoft (2006)
18. di Milano, P.: *A tutorial on clustering algorithms*. Home Polimi (2009)
19. Cohn, D.: *Mixtures of Gaussians*. School of Computer Science Carnegie Mellon University (1996)
20. Moore, A.W.: *Clustering with Gaussian Mixtures*. School of Computer Science. Carnegie Mellon University (2004)
21. Orjuela, A., Calôba, L.: Clasificación de Movimientos en Extremidades Usando Redes Neuronales: I. Proceso Supervisado. In: *21° Congresso Brasileiro em Engenharia Biomédicas* (2008)

Application Possibilities of Hardware Implemented Hybrid Neural Networks to Support Independent Life of Elderly People

Stefan Oniga¹ and Petrica Pop-Sitar²

¹ Technical University of Cluj-Napoca, North University Center Baia Mare,
Faculty of Engineering, 430033 Baia Mare, Romania
stefan.oniga@ubm.ro

² Technical University of Cluj-Napoca, North University Center Baia Mare,
Faculty of Sciences, 430122 Baia Mare, Romania
petrica.pop@ubm.ro

Abstract. With the explosion of new information and communication technologies and new devices, these offer increased opportunities to support everyday life but also have increased the requirements on the expected properties such as adaptability to user needs, behavior and particularities. These requirements are even more needed if the user is elderly, disabled or children. Properties like adaptability or learning capability, self-organization can be ensured by using interfaces that copies biological behavior. Thus, uses of Hybrid Artificial Intelligent Systems can represent key solution for obtaining adaptive interfaces and systems. Modeling such complex systems is some time to computational intensive that's why we have proposed the use of hardware implemented neural networks. Using Field Programmable Gate Arrays (FPGA) for hardware implementation allows parallel implementation of neurons increasing the processing speed. This paper aims to present the method developed by the authors for implementing artificial neural networks, the results obtained and the possibility of use in some applications experimented by the authors support independent life of elderly people.

Keywords: hybrid neural networks, behavior-finding, adaptive systems, hand position recognition, human activity recognition.

1 Introduction

Many today's world real systems have a complex structure and modeling such systems is often impossible using computers due to computational limitations. A possible approach is to describe the behavior of the system as a holistic model using hardware implemented neural networks [1]. This is even more obvious when we try to recognize patterns to model human activity or behavior like in:

- hand posture recognition [2], [3], [4]
- activity pattern recognition
- health state pattern recognition

There are many studies made for hand posture recognition [2], [3], [4], activity recognition based on image recognition [5], wearable body sensors [6], [7], PDAs or smart phones [8], [9], etc. Also there are studies regarding pattern recognition classification algorithms, most used algorithms are Decision Trees, Naïve Bayes, Support Vector Machine, k-Nearest Neighbor [10], [11]. Implementation can be done using smartphones or in case of supervised classification algorithm that needs intensive computation the implementation is done in servers. Also there are some implementations of neural networks classifiers in phones or server [12], [13], [14]. But only a few researches like ours have been made regarding activity recognition using hardware implemented neural networks [10].

The new ICT based devices offer increased opportunities to support everyday life but also have increased the expectations regarding properties such as adaptability to user needs, behavior and particularities. These requirements are even more needed if the user is elderly, disabled or children. Properties like adaptability or learning capability, self-organization can be ensured by using interfaces that copies biological behavior. Thus, uses of Artificial Intelligent Systems can represent key solution for obtaining adaptive interfaces and systems.

Modeling such complex systems is some time to computational intensive that's why we propose the use of hardware implemented neural networks. Using Field Programmable Gate Arrays (FPGA) for hardware implementation allows parallel implementation of neurons increasing the processing speed.

Among application possibilities of hardware implemented neural networks models are in complex systems for:

- posture/gesture commands
- intelligent interfaces like adaptive interfaces for PC, smart home, robots, etc.
- assistive robots
- artificial nose

This paper first presents shortly the method of hardware implementation of neural networks, a short overview of implemented neural networks and their applications. The focus of the paper is on hybrid neural networks modeling and their possible applications in everyday life support and independent living assistance of elderly or persons with disabilities. Two applications are presented: hand postures recognition and human activity recognition which demonstrates the capacity of hardware implemented neural networks using our method to learn and recognize patterns. They are part of a project aiming to develop ICT tools for smart homes and assisted living for elders. Using hardware implemented neural network we can develop intelligent system there is no need for a computer. In this way the resulting system is smaller and cheaper.

2 Method

One of the authors (Oniga) has developed earlier and applied an original method for hardware implementation of artificial neural networks (ANN) as was presented in [15]. We used System Generator extension for Simulink/Matlab created by Xilinx Inc. for designing high-performance DSP systems using FPGAs. With this high-level tool we developed a library that can be used for rapid prototyping of ANNs. The ANN can be trained either offline or online. The first approach that we used is to train the network in Matlab with Neural Network Toolbox, to save the weights in a file and then load them automatically in hardware weight memories for hardware implementation. Another approach is to use on-chip learning. Both methods have advantages and drawbacks. For the former one is no need for computer and Matlab to train the ANNs, the training stage could take only a few minutes, and the ANNs could be easily adapted to new input conditions. But the hardware implementation of some training algorithm could be difficult, and the space occupied by hardware needed to implement the learning phase leads to space limitation for the circuit that implements the NN [16]. The space limitation is always an issue in HW implementation of the ANNs because of the lack of a great number of hardware multipliers. This is why we have opted in most cases for offline training of the network. A new ANN can be designed instantiating, parameterizing and connecting together the blocks from our library and generic building blocks. In this way different types of ANNs can be implemented, choosing the best suitable network for the given application.

2.1 Neurons

For implementation of the neuron can be used for example the well known McCulloch-Pitts model of the neuron, presented in Fig. 1. Central element of the model is the summing junction having as inputs the weighted inputs of the neuron. It could be modeled using a hardware multiplier and an accumulator. The hardware multiplier blocks existing in Xilinx FPGAs can be used for an efficient implementation of the neuron multiplier. We used one multiplier per neuron implementing in this way a neuron parallelism. Fig.2 shows the same neuron modeled using hardware blocks from our library.

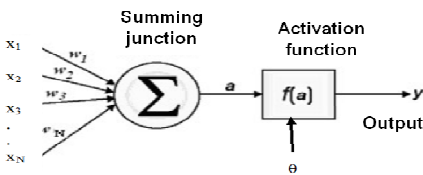


Fig. 1. McCulloch-Pitts model of the neuron

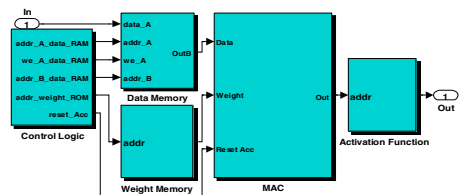


Fig. 2. Neuron block hardware model

$$y(x) = f(a - \theta) = f\left(\sum_{i=1}^N w_i x_i - \theta\right) \tag{1}$$

2.2 ANN Implementation Using System Generator

The ANN library developed using System generator tool is presented in Fig.3. It contains all necessary blocks for rapid prototyping of ANN (MAC blocks, activation function blocks, weight memory block, and control logic block). A new ANN can be designed instantiating, parameterizing and connecting together the newly created blocks and generic building blocks. In this way different type of ANNs can be implemented, choosing the best suitable for the given application.

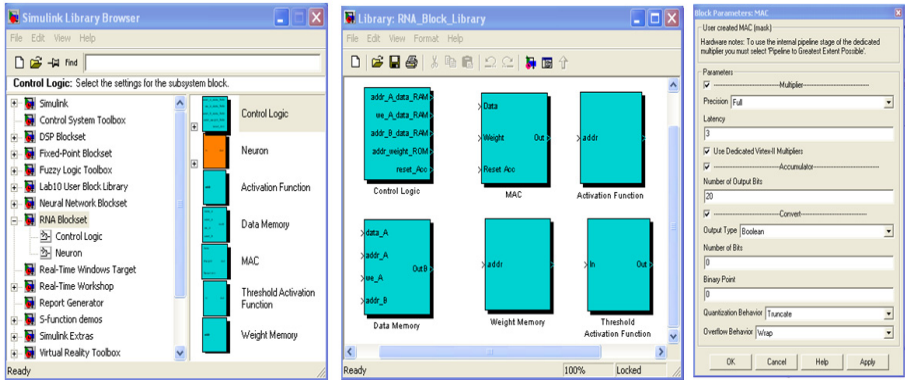


Fig. 3. ANN library and blocks with parameterizing GUI

With the method described above we have designed several types of ANNs such as Feed Forward Back Propagation (FF BP) network trained using Hebbian or Levenberg-Marquardt algorithm [17,18, 19], Competitive network [20, 21], Self Organizing Maps (SOM) network [22, 23], and also we implemented some hybrid networks.

Using these ANNs we designed many systems that exploit their learning capability and the adaptive behavior. Between tested applications are different pattern recognition systems for hand gesture recognition [4], artificial olfaction system [23, 24], intelligent Human-Machine Interface [2], smart sensors, smart devices [17], etc.

3 Hybrid Neural Networks Application in Pattern Recognition

As mentioned earlier one of the possible applications of the ANN is the pattern recognition task. This feature is very useful in applications related to support independent life of elderly people like in: hand posture recognition, activity pattern recognition, and health state pattern recognition. In this way the system could adapt to elderly or persons with disabilities and special needs. Next we present two applications developed for hand postures and activity recognition using hardware implemented neural networks.

3.1 Hand Postures Recognition System Implementation Using Hybrid Neural Networks

In a previous work we developed a system for hand posture recognition based on data acquired from a sensorial data glove with optical fiber bend sensors. After we tried several neural networks architectures we concluded that the best performing was a hybrid neural network composed of a FF-BP and a competitive network.

Next architecture e experimented is a hybrid, two levels architecture composed from:

- Preprocessing FF-BP ANN
- Gesture classifying Competitive ANN

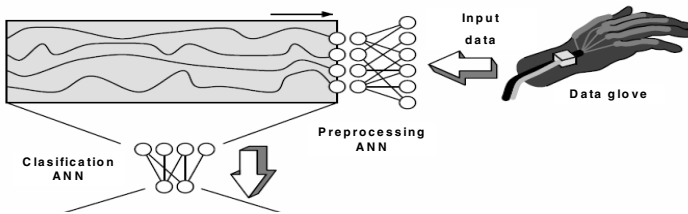


Fig. 4. Architecture of the hand posture recognition system using hybrid neural networks

Implemented Function

First layer implements the function described in Equation 2:

$$y_j^{(1)}(x) = f \left(\sum_{i=1}^M w_i^{(1)} x_i^{(1)} - \theta^{(1)} \right) \tag{2}$$

where $j = 1, 2, \dots, N_1$, represents neurons on first network, ($N_1=7$), and $i=1, 2, \dots, M$, ($M=7$) is the number of inputs of the neurons. In case of linear activation function:

$$y_j^{(1)}(x) = \sum_{i=1}^M w_i^{(1)} x_i^{(1)} - \theta^{(1)} \tag{3}$$

Net output of the neurons of the second network is:

$$net_k = \sqrt{\sum_j^{N_1} [y_j^{(1)} - w_{kj}^{(2)}]^2} \tag{4}$$

where $k = 1, 2, \dots, N_2$, ($N_2=15$) represents neurons of second network. Simplifying Equation 4 to an equivalent hardware friendly form we obtained:

$$net_k = \sum_j^{N_1} [y_j^{(1)} - w_{kj}^{(2)}]^2 \tag{5}$$

So the final form of the net output is given by Equation 6:

$$net_k = \sum_j^{N_1} \left[\left(\sum_{i=1}^M w_i^{(1)} x_i^{(1)} - \theta \right) - w_{kj}^{(2)} \right]^2 \tag{6}$$

The activation function of the neurons on second network is the competitive activation function.

Fig. 5 shows architecture of the hybrid neural network used for posture recognition.

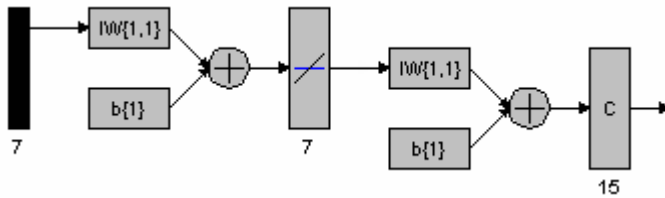


Fig. 5. Architecture of hybrid neural network

In our current work we made experiments to identify hand posture recognition using acceleration sensor data.

Hand Postures Definition

We have defined some common hand postures aiming to recognize this postures using ANN. The defined postures are presented in next table 1.

Table 1. Hand Postures

1. front	2. up	3. down
4. front left	5. up left	6. down left
7. front right	8. up right	9. down right

Experimental Setup

We have developed an acquisition setup composed from the Chronos data watch from Texas Instruments together with its access point connected to USB port of the computer. The Chronos watch is equipped with sensors: temperature, 3-axis acceleration, pressure & altitude sensor, heart rate (with attached chest belt). Also we developed a Python program that sends commands to the access point to establish the communication and to require acceleration data. The acceleration data are saved on the computer.

Training and Test Data Acquisition for Postures Recognition

Acceleration data could be sampled at the desired frequency. We determined that a good sampling frequency for hand postures is 10 Hz. For each position we acquired 200 samples that are used for training and testing the artificial neural networks. The acquired data requires some processing that is made with a Matlab m file.

ANN Design

Is done using Neural Network toolbox, but it could be done also using a Matlab code that is capable to train the network. We must try several NN architecture and several training rules in order identify the best suitable network, which supplies the best recognition rate. This network will be implemented in hardware.

Results obtained using a FF-BP trained with Levenberg-Marquand algorithm with 10 neurons on the hidden layer and 9 neurons on the output layer corresponding to 9 postures to be recognized, is presented in Fig.4. Transfer functions are sigmoid for the hidden layer and linear for output layer. Network was simulated using 200 samples for each postures so a total of 1800 samples. It could be observed that recognition rate is very good for first tree postures and bad for postures 4-6. Total number of errors is 512 meaning only 3.16%, so recognition rate of 96.84%. Performance function given by mean square error is 0.0231.

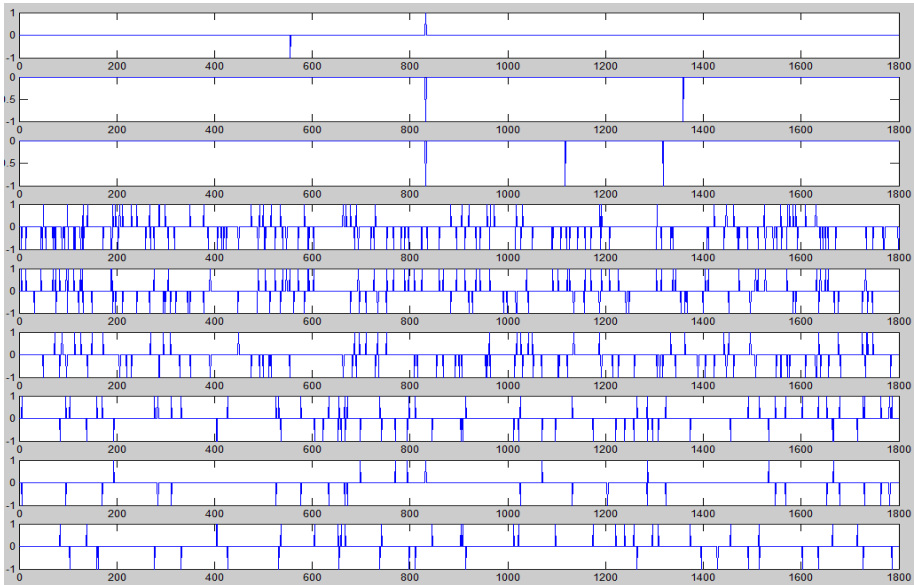


Fig. 6. Errors of the posture recognition using FF-BP

3.2 Human Activity Recognition

Our current work is related to activity recognition based on acceleration data. We made experiments to recognize some common body positions like sitting, prone, supine, left lateral recumbent, right lateral recumbent (Fig. 7).

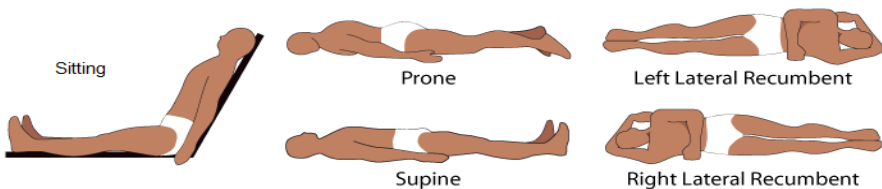


Fig. 7. Body posture definition

Monitoring acceleration data one can detect the body positions. Fig. 8 shows acceleration data for the 5 body positions defined above obtained using TI Chronos watch.

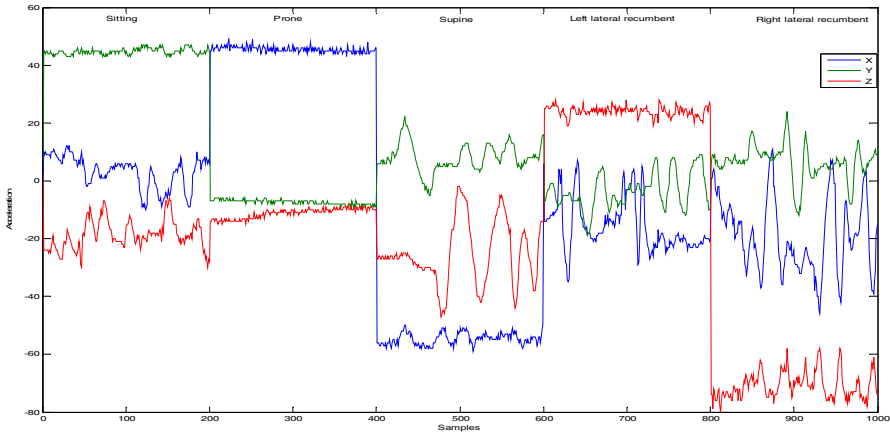


Fig. 8. Acceleration data for 5 body positions

Using this data we have trained a FF_BP ANN ANN with 10 neurons on the hidden layer and 5 neurons on the output layer. Simulation results of this network are presented in Fig. 9. Total number of errors is 2 meaning only 0.04%, so recognition rate of 99.96%. Performance function given by mean square error is $3.6747e-004$.

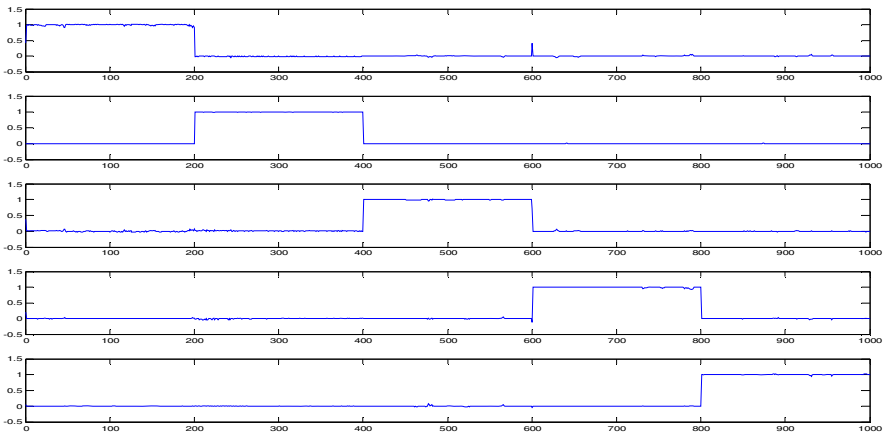


Fig. 9. Simulation results showing neurons outputs with the recognized position

4 Conclusions

We obtained very good recognition rate for a part of the hand postures. The explanation for errors on some postures is that we cannot discern between hand positions using only the acceleration sensor. In fact the acceleration data are almost identical for positions 2, 5 and 8 respectively for 3, 6 and 9. Properly choosing the hand positions we could recognize 6 postures with almost 100% rate. In order to recognize more postures we need data from other sensors like magnetometer and gyroscope.

It is possible to extract features from acceleration signal and current body position recognition with trained neural networks. The normal body position could be recognized with a high precision rate using only an accelerometer. Our current work is related to activity recognition based on acceleration data. Monitoring acceleration data one can detect activity types: standing, walking, running, sitting, falling and other. For a better recognition of activity type we will use more sensors and sensorial fusion. For example combining data from different sensors (temperature, acceleration, EKG, heart rate) it is easier to identify the state of a monitored patient. Another possible improvement could be obtained using fuzzy logic rules.

There are many application possibilities of hardware implemented ANN for ICT devices development used in elderly or people with disabilities, everyday life assistance. We have presented results obtained so far in two applications that we have developed. The first is the hand posture recognition and the second is the body position detection. In our future work related to activity detection we will use more sensors and sensor data fusion and fuzzy logic rules.

We proposed, tested and proved that neural network could be implemented in FPGAs instead of using computers and could deliver very good recognition rate. This leads to a smaller chipper and intelligent assistive device.

Acknowledgement. This work was supported by a grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-II-RU-TE-2011-3-0113.

References

1. Tisan, A., et al.: Holistic modeling, design and optimal digital control of a combined renewable power system. In: 2010 IEEE International Symposium on Industrial Electronics, ISIE 2010, Bari, Italy, July 4-7, pp. 2733–2738 (2010)
2. Oniga, S., Vegh, J., Orha, I.: Intelligent Human-Machine Interface Using Hand Gestures Recognition. In: 2012 IEEE International Conference on Automation Quality and Testing Robotics, AQTR, pp. 559–563 (2012)
3. Ghotkar, A.S., et al.: Hand gesture recognition for indian sign language. In: International Conference on Computer Communication and Informatics, pp. 1–4. IEEE (2012)
4. Oniga, S., Tisan, A., Mic, D., Buchman, A., Vida, A.: Hand Postures Recognition System Using Artificial Neural Networks Implemented in FPGA. In: 30th International Spring Seminar on Electronics Technology, Cluj-Napoca, Romania, May 9-13, pp. 507–512 (2007)
5. Bandouch, J., Jenkins, O.C., Beetz, M.: A Self-Training Approach for Visual Tracking and Recognition of Complex Human Activity Patterns. *International Journal of Computer Vision* 99(2), 166–189 (2012)
6. Becher, K., Figueiredo, C.P., Mühle, C., Ruff, R., Mendes, P.M., Hoffmann, K.-P.: Design and realization of a wireless sensor gateway for health monitoring. In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), August 31-September 4, pp. 374–377 (2010)
7. Zhang, Y., Markovic, S., Sapir, I., Wagenaar, R.C., Little, T.D.: Continuous functional activity monitoring based on wearable tri-axial accelerometer and gyroscope. In: 2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), May 23-26, pp. 370–373 (2011)

8. Wu, W., Dasgupta, S., Ramirez, E.E., Peterson, C., Norman, G.J.: Classification Accuracies of Physical Activities Using Smartphone Motion Sensors. *J. Med. Internet. Res.* 14(5), e130 (2012)
9. Mukhopadhyay, S.C., Postolache, O.A. (eds.): *Pervasive and Mobile Sensing and Computing for Healthcare Technological and Social Issues*. Springer, Heidelberg (2013)
10. bin Abdullah, M.F.A., et al.: Classification algorithms in human activity recognition using smartphones. *Proceedings of World Academy of Science, Engineering and Technology* (68) (2012)
11. Kouris, I., Koutsouris, D.: A comparative study of pattern recognition classifiers to predict physical activities using smartphones and wearable body sensors. *Technology and Health Care* 20(4), 263–275 (2012)
12. Györfbíró, N., Fábíán, Á., Hományi, G.: An Activity Recognition System for Mobile Phones. *Mobile Networks and Applications* 14(1), 82–91 (2008)
13. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity Recognition using Cell Phone Accelerometers. *Human Factors* 12(2), 74–82 (2010)
14. Khan, A.M., Lee, Y., Lee, S.Y.: Human Activity Recognition via an Accelerometer-Enabled-Smartphone Using Kernel Discriminant Analysis. In: *Proceedings of 5th International Conference on Future Information Technology*, pp. 1–6 (2010)
15. Oniga, S.: A New Method for FPGA Implementation of Artificial Neural Network Used in Smart Devices. In: *International Computer Science Conference microCAD 2005*, Miskolc, Hungary, pp. 31–36 (March 2005)
16. Omondí, A.R., Rajapakse, J.C.: *FPGA Implementations of Neural Networks*. Springer (2006)
17. Oniga, S., et al.: FPGA Implementation of Feed-Forward Neural Networks for Smart Devices Development. In: *International Symposium on Signals, Circuits and Systems*, Iasi, Romania, July 9–10, pp. 401–404 (2009)
18. Oniga, S., Tisan, A., Mic, D., Buchman, A., Vida, A.: Optimizing FPGA Implementation of Feed-Forward Neural Networks. In: *Proceedings of the 11th International Conference on Optimization of Electrical and Electronic Equipment*, Brasov, Romania, pp. 31–36 (2008)
19. Tisan, A., Oniga, S., Gavrincea, C.: Hardware implementation of a MLP network with on-chip learning. In: *Proceedings of the 5th WSEAS Int. Conf. on Data Networks, Communications & Computers*, Bucharest, Romania, pp. 162–167 (2006)
20. Oniga, S., Tisan, A., Buchman, A., Lung, C.: Hardware Implementation of Simple Competitive Artificial Neural Networks with Neuron Parallelism. In: *Proceedings of Regional Conference on Embedded and Ambient Systems*, Budapest, Hungary, pp. 27–32 (2007)
21. Oniga, S., et al.: Hardware implementation of simple competitive neural networks with layer parallelism. In: *International Symposium for Design and Technology of Electronic Packages, SIITME 2007*, Baia Mare, Romania, September 20–23, pp. 193–198 (2007)
22. Tisan, A., Cirstea, M.: SOM neural network design – A new Simulink library based approach targeting FPGA implementation. *Math. Comput. Simul.* (2012), <http://dx.doi.org/10.1016/j.matcom.2012.05.006>
23. Tisan, A., Oniga, S., Gavrincea, C., Buchman, A.: FPGA implementation of a Self-organized map with on-chip learning. In: *Proceedings of the 11th International Conference on Optimization of Electrical and Electronic Equipment*, OPTIM 2008, Brasov, Romania, pp. 81–86 (2008)
24. Tisan, A., Cirstea, M., Oniga, S., Buchman, A.: Artificial olfaction system with hardware on-chip learning neural networks. In: *12th International Conference on Optimization of Electrical and Electronic Equipment*, OPTIM 2010, Brasov, Romania, pp. 884–889 (2010)

Multi-agent Reactive Planning for Solving Plan Failures

César Guzmán-Alvarez¹, Pablo Castejon¹, Eva Onaindia¹, and Jeremy Frank²

¹ Universitat Politècnica de València, Camino de Vera s/n, Valencia, Spain

² NASA Ames Research Center, Moffet Field, USA

Abstract. In this paper we present a multi-agent reactive planning mechanism for recovering from plan failures with the help of multiple agents. Our contribution is twofold: a proposal of a dynamic execution architecture embedded into a more general multi-agent planning framework, and a mechanism based on state-transition systems that allows execution agents to reactively and cooperatively attend a plan failure during execution. Specifically, we propose a flexible dynamic execution architecture that allows agents to find solutions for a successful plan execution during a plan failure.

Keywords: reactive planner, multi-agent planner, coordination, execution.

1 Introduction

Most planning-and-execution applications rely on single-agent architectures that include the functionalities required for a continuous planning, namely sensing the state, generating the problem at hand, planning, executing the plan, monitoring the execution for failures, and replanning; for example, space and robotics applications of platforms as Mapgen [1], APSI [5], PRS [8], or IxTeT [9]. Typically, these architectures incorporate a deliberative component augmented with reactive behaviors [14], unify deliberation and execution under a single planning technology and model representation [2] or maintain independent modules for planning and execution in an integrated way [10].

Multi-agent planning (MAP) systems are viewed as extensions of planning-and-execution single-agent architectures for cooperative distributed problem solving [16,11,12]. One common characteristic of these architectures is that the multi-agent infrastructure is specifically used for supporting the deliberative machinery (planning) whereas the need for reactive mechanisms (plan execution) is basically relegated to the individual agent level. Thus, when an executor agent encounters a failure during plan execution it either resorts to a centralized manager that sends messages to the planning agents requesting a solution [16]; it accommodates some sort of reactivity by having task assessors that only abstractly plan how to accomplish the failed task [12]; or the executor agent is a Beliefs, Desires and Intentions (BDI) agent that exhibits a reactive behavior and responds to a plan failure by consulting a plan library of predefined, static plans [13]. In any case, almost all of the MAP architectures rely on cooperative behaviors for the construction of plans but execution failures are individually attended by each agent.

Reactive planning architectures have been largely investigated in the area of automated planning. The first approaches to reactive planning exploited abstraction as a

means to implement quick response mechanisms, like the Procedural Reasoning System (PRS) [8], a framework for symbolic reactive control systems in dynamic environments, or the Reactive Action Package (RAP) [7], a system designed for the reactive execution of symbolic plans. The usage of hierarchical control structures [3] or semi-reactive architectures [9], which provide rough plans when a quick response is required in the presence of unforeseen events, are also very popular in reactive planning. However, none of the reactive frameworks embedded in MAP systems have ever exploited the idea of cooperatively solving a plan failure at execution time by using the reactive capabilities of the agents in the system. The problem here lies in the difficulty of merging the reactive plan representation of multiple agents and reactively responding to an agent's request.

In this paper, we present a first approach to reactively and cooperatively solve a plan failure in a MAP system. Our work builds upon PELEA [10], a component-based single-agent architecture able to perform planning, execution, monitoring and repairing in an integrated way, and PLANINTERACTION¹, a multi-agent planning architecture that integrates PELEA agents into a multi-agent system. Within the PLANINTERACTION platform, we propose a dynamic execution architecture and a recovery mechanism based on state-transition systems that allow executor agents to quickly respond to unexpected events before resorting to a computationally expensive replanning solution.

This paper is organized as follows. Next section provides the main features of PELEA, PLANINTERACTION, and presents the Dynamic Execution Architecture embedded in PLANINTERACTION. Section 3 introduces the state-transition system used by the reactive planner and section 4 presents an example of application. Finally, last section concludes and presents our future research lines.

2 PlanInteraction: An Architecture for Multi-agent Plan Interaction

PLANINTERACTION² is a multi-agent planning-and-execution architecture, which allows agents to autonomously perform science targets, execute a set of tasks in a simulated or real world, monitor the plan execution attending to potential discrepancies, and take decisions for repairing or replanning in case of a plan failure.

PLANINTERACTION is built upon PELEA [10], a Planning, Execution and LEarning Architecture for a single-agent. PELEA provides an agent with onboard capabilities to generate, execute, and monitor a plan. It also provides an agent with learning capabilities. The main components of PELEA that are used in PLANINTERACTION are:

- *Execution module (EX)*. This is the starting point of the architecture. The *EX* captures as input a planning task, which current state is read from the environment through the sensors. The *EX* module is thus in charge of reading and communicating the current state to the rest of modules as well as executing the tasks in the environment.

¹ <http://servergrps.dsic.upv.es/planinteraction/>

² This work is supported by project TIN2011-27652-C03-01

- *Monitoring module (MO)*. Besides the current state, the *EX* also sends the *MO* the planning task to solve. The *MO* calls a deliberative planner in order to obtain a plan (see below) and, once obtained, it sends the actions to execute to the *EX*. The *EX* reports the *MO* the state resulting from executing each action and the *MO* performs the plan monitoring process, i.e. it checks whether the received state matches the expected state or not and determines the existence or lack of a plan failure.
- *Deliberative Planner module (DP)*. The *DP* receives a planning task and generates a plan for the task. This module is also invoked when it is necessary to fix (repair or replan) a plan. The planner is also responsible for selecting the variables that the module *MO* must check during the plan execution. Any state-of-the-art planner can be used as the *DP* module.

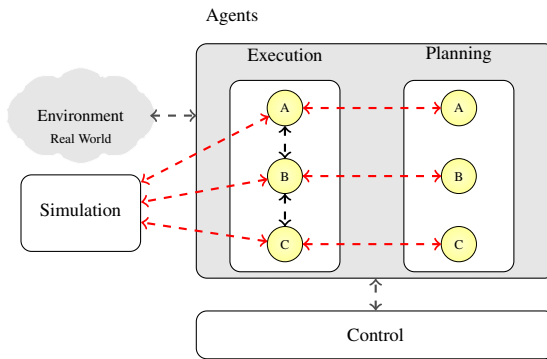


Fig. 1. Multi-Agent Plan Interaction Architecture

PLANINTERACTION is a flexible and domain-independent architecture implemented by integrating PELEA agents in an open MAP platform called MAGENTIX2 [15]. As we can see in Fig. 1, the architecture consists of three main modules:

- *Control* is responsible of registering agents in the system, initializing the internal clock, handling the problem information and controlling conditions for the system termination. The internal clock manages the time of all agents in the system.
- *Simulation* represents the simulated state of the world. Agents will be able to access the information in the simulated environment as well as to modify it through the execution of the actions in their plans.
- *Agents* comprises the set of agents of the problem. An agent in PLANINTERACTION represents any combination of the PELEA modules. The composition of modules in each agent depends on the problem specification and agent's capabilities. Thus, we can have the three modules (*EX*, *MO* and *DP*) embedded in a single PELEA-like agent; we can also opt for creating planning agents that only comprise the *DP* module, thus providing agents with capabilities for planning and repairing plans, and execution agents that comprise the *EX* and *MO* modules, with capabilities for tracing plan execution and plan monitoring (this is the configuration shown in Fig. 1). In some applications, we might even want to have several different EX modules but a single MO; for example, a robotics application where one monitor controls the operations of several robots moving in a shared space.

The particular architecture configuration we have chosen for our purposes is shown in Fig. 1. The simulation module is not required in case we are working in a real-world scenario. Each execution agent comprises the modules *EX* and *MO*, and each planning agent contains a *DP* module. Specifically, in the problem shown in section 4 we have three parties, two rovers (A, and B) and one spacecraft (C), each one containing a planning agent and an execution agent.

In this type of configuration, we distinguish three different coordination levels: 1) *Planning-Planning Coordination*, between planning agents when they have to jointly generate a solution plan for a particular task; 2) *Execution-Execution Coordination* focuses on the coordination between execution agents when they attempt to resolve a plan failure at execution time; 3) *Execution-Planning Coordination* takes place when execution agents are not capable of reaching a solution during the execution-execution coordination and so they have to resort to their planning agents so as to find a new plan for solving the task. In this paper, we specifically focus on execution-execution coordination.

2.1 Dynamic Execution Architecture

In this subsection, we present the multi-agent Dynamic Execution Architecture (DEA) we have implemented within the PLANINTERACTION framework. DEA is particularly devoted to implement the execution-execution coordination referred above. Our goal is to come up with DEA in which execution agents gather together at execution time for repairing some failure that happened in the environment before resorting to a more computationally expensive planning-execution coordination.

DEA is composed of three modules (Fig. 2 left): a *MO* module, a Reactive Planner (*RP*), and an *EX* module. The control flow of the architecture (Fig. 2) begins when the planning agent sends the plan and the parameters to be monitored (*monitor-parameters*) to the *MO* module of the execution agent. The *MO* sends the plan to the *RP*, which

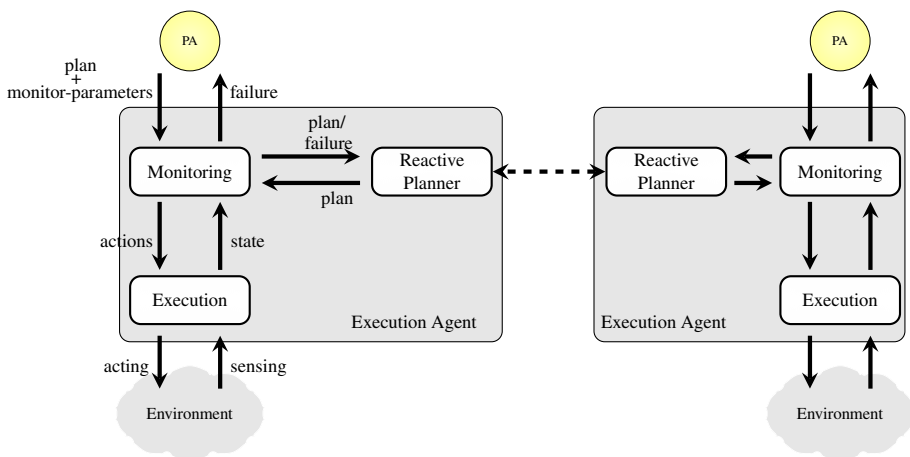


Fig. 2. Multi-Agent Dynamic Execution Architecture

transforms the plan into a set of reactive state-transition systems (explained in the next section) that will be later used for a reactive plan fixing. Meanwhile, the *MO* sends a set of executable (*actions*) to the *EX*, which executes them in the specified order (*acting*), senses the dynamic part of the state from the environment (*sensing*), and sends it to the *MO*. The *MO* receives the information from the sensors (*state*) and checks the parameter values before sending the next action to execute. If a discrepancy in the value of a variable is found, the anomaly (*failure*) is reported by the *MO* to the *RP*, which uses the stored reactive structure for a rapid intervention. If the *RP* finds a solution, it sends a new plan to the *MO*, which in turn sends the next action to the *EX*. In case the *RP* is not able to fix the problem, it can request other agents for help. This implies the activation of a communication protocol that performs the execution-execution coordination. If the other agents cannot find a solution, then the *RP* informs the *MO*, which reports the plan failure to the planning agent for that it repairs the plan or find a new executable plan.

3 Reactive Planner

We briefly summarize some basic concepts of planning that we will need to explain the *RP* module. It is important to note that we work at the same abstraction level both within a planning and an execution agent. While many architectures translate high-level planning specifications into low-level execution structures, we keep the same level of abstraction. This is by no means a loss of generality as it has no impact at all in the design of the reactive planning module.

In classical planning, a planning task of an agent is defined as a tuple $\langle I, A, G \rangle$, where I represents the agent's initial state of the world, G is a partial world state that represents the goals that the agent wants to achieve and A is the set of actions of the agent. In our setting, we assume we have a set of entities or domain agents, each one with its particular planning task to solve. Agents share the initial state I and, possibly, some of the actions in A , depending on the agents' skills; however, each agent has its specific G to accomplish.

A state of the world is modeled through a finite set of state variables V , each associated to a finite domain D_v of mutually exclusive values. A *fluent* is a tuple $\langle v, d \rangle$, which indicates that the variable $v \in V$ takes the value $d \in D_v$. Therefore, a world state s is defined as a set of fluents.

An agent's action is a transition function $a \in A$ that when applied to world state s gives rise to a new state s' . Specifically, an action a is defined as tuple $a = \langle pre(a), eff(a) \rangle$, where $pre(a)$ is a finite set of fluents that represents the preconditions of a , the fluents that must hold in s in order to apply a in s . And $eff(a)$ is a finite set of operations that change the value of fluents. An operation of the form $(v = d)$ adds a fluent $\langle v, d \rangle$ to state s' and also removes fluents of the form $\langle v, d' \rangle$ such that $d' \neq d$ in state s' . We will denote by $eff(a)^+$ the fluents added to s' and by $eff(a)^-$ the fluents removed in s' .

An agent's plan is defined as $\pi = \langle a_1, a_2, \dots, a_n \rangle$, a sequence of actions that solves its planning task. This way, action a_1 is applied in state I resulting in a new state, say s_1 , then a_2 is applied in s_1 resulting in a new state and so on. Given a world state s and an action a , the result of executing a in s is $result(s, \langle a \rangle) := s \setminus eff(a)^- \cup eff(a)^+$ if the action is applicable in s , i.e., $pre(a) \subseteq s$. Otherwise, $result(s, \langle a \rangle)$ is undefined.

The result of executing π in a state is recursively defined by $result(s, \langle a_1, \dots, a_n \rangle) := result(result(s, \langle a_1, \dots, a_{n-1} \rangle), a_n)$, and $result(s, \langle \rangle) = s$.

When the *MO* module of an execution agent receives a plan $\pi = \langle a_1, a_2, \dots, a_n \rangle$, the *MO* sends action a_1 to execute and, simultaneously, sends π to the *RP* module. The reactive planner converts the plan π into a reactive structure that allows it to attend future failures during the plan execution. Specifically, the *RP* builds a state-transition system [4] that represents the plan π and extends this basic representational structure by adding new world states (along with their corresponding transitions) that denote failed states that are likely to be reached during the plan execution. State-transition systems are commonly used in model-checking planning approaches [6].

Definition 1. A state-transition system consists of a set of (world) states and transitions between states, which are labeled with actions from the set A . A state-transition system T is defined as a 4-tuple $T = \langle F, S, A, R \rangle$, where:

- F is a finite set of fluents
- $S \subseteq 2^F$ is a finite set of states
- A is a finite set of actions over S
- R is the state-transition function $R: S \times A \rightarrow S$ that represents a transition between two states labeled with an action from A .

Given a plan $\pi = \langle a_1, a_2, \dots, a_n \rangle$ of an execution agent, we will denote the sequence of states generated through a successful execution of π as $S = \langle s_0, s_1, s_2, \dots, s_{n-1}, s_G \rangle$, where:

- $s_0 = I$ is the initial state, and s_G is the final state such that $G \in s_G$
- $result(s_0, a_1) = s_1, result(s_1, a_2) = s_2, \dots, result(s_{n-1}, a_n) = s_G$

The plan π and the sequence of states S traversed by a successful plan execution define the *basic T* of an execution agent. Particularly, S is the sequence of states; F is the set of fluents contained in all of the states in S ; A is the set of actions in π ; and R is the *result* function. In the following, we will refer to a basic state-transition system as a tuple $T = \langle \pi, S \rangle$.

3.1 Extending the Basic State-Transition System

Once the reactive planner builds T from the plan of an execution agent, the next step is to extend it by including new states and transitions in T . We can distinguish two main situations when a plan failure occurs. The **first situation** is that the failure in the action execution makes the agent remain exactly at the same state that it was initially. In other words, this situation occurs when $result(s_i, a_{i+1}) = s_i$. In this case, the agent finds itself in the same state and an alternative course of actions from s_i is necessary, if possible, to reach s_{i+1} . This situation is usually due to a malfunction in the action execution that leaves the world unaffected. The **second situation** typically arises when exogenous events occur in the environment. In this case, after the action execution, the agent is neither at s_i nor s_{i+1} but in a different state, where at least one of the fluents that model the resulting world state does not have the expected value.

Given a basic state-transition system $T = \langle \pi, S \rangle$ of an execution agent, T is extended to account for the first situation as follows: for each state $s_i \in S$, we search for alternative transition paths that connect s_i to another state in S ; that is, we search for applicable actions in s_i (actions such that $pre(a) \subseteq s_i$), other than a_{i+1} , that lead to any forward recovery state $s_{i+1}, s_{i+2}, \dots, s_G$. This process is performed through a forward state-space search. On the other hand, we might consider there is only one desirable recovery state to reach: that is, the following state in the sequence. If the *MO* module detects an error when trying to reach state s_i , it is likely the *MO* would request a plan to reach solely s_{i+1} as this solution would permit then to continue with the execution of the next action in π . This is so because in reactive planning a quick response that allows to continue with the plan execution is preferable over a more time-consuming solution. The consideration of exogenous events in $T = \langle \pi, S \rangle$ is addressed as follows:

- 1) For each pair (s_i, a_{i+1}) such that $s_i \in S$, $a_{i+1} \in \pi$ and a_{i+1} is applicable in s_i , we create a list $L = \{\langle v, d \rangle\}$ that contains the fluents that appear in $pre(a_{i+1})$.
- 2) Let v be a variable of a fluent $\langle v, d \rangle$ in L whose domain is $D_v = \{d_1, \dots, d_m\}$; for each value $d_i \neq d$, we create a new state $s' = s \setminus \langle v, d \rangle \cup \langle v, d_i \rangle$. This operation is repeated for the variables that appear in all fluents in L .
- 3) We connect the new states generated in 2) to another state in S by following the same procedure explained above.

Therefore, we consider all possible contingencies in the value of the variables that appear in the preconditions of the actions of π . It is also possible that none of the states in S are reachable from a state generated due to an exogenous event; that is, there is no transition path from the new state to any of the states in S .

An extended state-transition system T' is a tuple $T' = \langle \pi, S, A', S' \rangle$, where π and S are the components of the basic state-transition system T , and A' and S' are the added transitions and states, respectively.

3.2 Fixing a Plan Failure

The list *monitor-parameters* that the *MO* module receives from the planning agent (see Fig. 2) contains elements of the form $\langle v, d, t_s, t_e \rangle$, where v is the variable, d is the expected value for v and $[t_s, t_e]$ is the time interval during which v is expected to take value d . When the *MO* receives the *state* from the *EX* after the execution of an action, the *MO* checks whether all the fluents contained in the *state* match the items in the *monitor-parameters* list or not. If not, the *MO* calls the *RP* to inform about a plan failure, passing the *RP* the current world *state* received from the sensors of the *EX* (we will call this state s_{cur} in the following), the recovery state to reach (s_{rec}) and the particular set of fluents that need to be repaired; i.e, the failed variables along with their expected values.

Assume $T' = \langle \pi, S, A', S' \rangle$ is the extended state-transition system defined in the *RP*. When the *RP* receives s_{cur} and s_{rec} from the *MO*, it performs the following operations: 1) find a state in $S \cup S'$ such that $s_{cur} \in S \cup S'$; 2) find a state in S such that $s_{rec} \in S$; 3) apply a simple version of the Dijkstra's algorithm to find a path from s_{cur} to s_{rec} . The state s_{rec} will always be a state from S as the objective is to take the execution back to a state from the original plan. The state s_{cur} will always be a state from either S

(first situation) or S' (second situation) since we consider all possible fluents that may appear in the state-transition system model. However, it might be the case there is not a transition path from s_{cur} to s_{rec} in whose case the *RP* will resort first to the other agents.

Multi-Agent Reactive Planner. The idea of cooperatively solving a plan failure at execution time requires to combine the state-transition systems of the agents involved in the system. Let EA_1 and EA_2 be two execution agents and $T_1 = \langle \pi_1, S_1, A'_1, S'_1 \rangle$ and $T_2 = \langle \pi_2, S_2, A'_2, S'_2 \rangle$ be their extended state-transition systems, respectively. First thing to note is that the planning tasks of the agents are different and so will be their sets of actions A_1 and A_2 and, equivalently, the set of fluents and states in T_1 and T_2 . However, EA_2 will be able to help EA_1 fix its plan failure if the fluent to be repaired belongs to the knowledge shared between both agents. The *shared data* by the entities of the problem is defined at planning time before the deliberative planner builds the plans for the agents. Assume $\langle v, d \rangle$ is the fluent to repair that the *RP* of EA_1 sends to the *RP* of EA_2 . EA_2 will match its s_{cur_2} in T_2 as well as the states in $S_2 \cup S'_2$ in which $\langle v, d \rangle$ holds. If a path from s_{cur_2} to any of the states holding $\langle v, d \rangle$ exists then EA_2 can actually help EA_1 . The final response will depend on how this transition path deviates from its plan π_2 and the cooperative behavior defined in the agents. Finally, if EA_2 cannot actually help EA_1 or is *not willing* to, EA_1 will call its planning agent for replanning the problem, i.e., find a new plan.

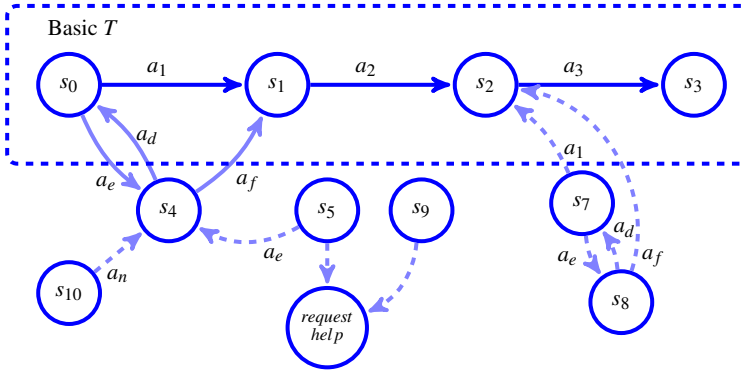
4 MARS Domain, an Example of Application

In this section, we will show how our reactive architecture works on a possible Mars Domain scenario. Space missions of NASA have rovers in Mars. When a plan failure occurs, rovers communicate with a control center on Earth for repairing the failure. NASA is interested in providing rovers with on-board reactive planning and execution capabilities, so that they can perform the reparation by themselves or with the help of other agents in a timely fashion and thus reducing the communication overhead with the Earth.

We present a simple example that shows the state-transition system of a rover for repairing future plan failures. Let's suppose we have a rover A , whose mission is to analyze rocks and communicate the results to a Lander L , which in turn sends the results to the Earth. There are three waypoints $\{w_1, w_2, w_3\}$ located on the surface; w_2 is the initial location of L and the rover A . Rover A has good maps to travel from w_1 to both w_2 and w_3 , and from waypoint w_2 to w_3 . The mission of the rover is to use the microscopic camera to analyze rocks in waypoint w_1 and communicate the results to L . The deliberative planner computes the plan $\pi = \langle a_1, a_2, a_3 \rangle$ for A (see Fig. 3 bottom).

Using the plan π , the *RP* module generates the basic transition system T , which consists of the states $S = \langle s_0, s_1, s_2, s_3 \rangle$, and transitions labelled with actions a_1, a_2, a_3 , as shown in Fig. 3.

Then, the *RP* extends T by considering the two situations explained in 3.1: those cases in which the rover remains in the same state after executing an action, and exogenous events. In the first situation, T is extended with state s_4 and transitions $A' = \langle a_e, a_d, a_f \rangle$. The transition a_e represents the action (navigate A w_2 w_3), which changes the fluent $\langle pos_A, w_2 \rangle$ to $\langle pos_A, w_3 \rangle$ and generates the new state s_4 . The transition a_f ,



$$\pi = \langle a_1:(navigate\ A\ w_2\ w_1), a_2:(sample\ rock\ A\ rover\ A\ store\ w_1), a_3:(comm\ Sample\ rock\ A\ L\ w_1\ w_2) \rangle$$

Fig. 3. State-transition system T for A

which is a path to the state s_1 , represents the action (navigate $A\ w_3\ w_1$), which changes the fluent $\langle pos_A, w_3 \rangle$ to $\langle pos_A, w_1 \rangle$.

T is now augmented with the states that may arise with the appearance of exogenous events. The states labeled as $s_5, s_7, s_8, s_9, s_{10}$ are incorporated to T along with the transitions shown in Fig. 3³. For example, s_5 represents a situation in which the rover finds the path from w_2 to w_1 is blocked. Then, s_5 will be the same state as s_0 except that the fluent $\langle path_{w_2w_1}, true \rangle$ is not present in s_5 . Since A can reach w_3 from s_5 , the transition a_e also connects s_5 to s_4 . Another example of exogenous event is s_9 , which represents the lander is not in w_2 , where it was supposed to be to communicate the results. As the position of L is unknown to rover A , it is not possible to reach a state of S from s_9 , so the only possible solution is to request for help to the other rovers or to the Earth. If, on the contrary, the rover A knew that L is in w_3 then a transition from s_9 to s_3 would appear in the state transition system.

As it can be observed, the information comprised in the state transition system T allows rover A to quickly find a solution to a failure because all the possible contingencies which can be modeled with the variables in the agent’s domain are considered in T . Additionally, it is easy to combine the information from two or more different state transition systems.

5 Conclusions and Future Works

In this paper, we have presented a first approach to a recovery mechanism from plan failures that makes use of state-transition systems as flexible reactive structures. Each agent comprises its own transition system that accommodates a subset of the possible failed states that the agent would encounter during the execution of its plan. Through this reactive structure, we can easily locate the failed state and find, if possible, a sequence of transitions that lead the agent to a recovery state. Additionally, agents can use

³ For simplicity, we do not show all the states that would be generated.

their state-transition systems to respond to agents' requests. In future works, we intend to exploit this research direction to create a conflict resolution mechanism for solving plan failures among multiple agents using reactive *teamworks* at execution time that work together in the accomplishment of a cooperative goal.

References

1. Ai-Chang, M., Bresina, J., Charest, L., Chase, A., Hsu, J.J., Jonsson, A., Kanefsky, B., Morris, P., Rajan, K., Yglesias, J., Chafin, B., Dias, W., Maldague, P.: MAPGEN: Mixed-initiative planning and scheduling for the Mars Exploration Rover mission. *IEEE Intelligent Systems* 19(1), 8–12 (2004)
2. Aschwanden, P., Baskaran, V., Bernardini, S., Fry, C., Moreno, M., Muscettola, N., Plaunt, C., Rijsman, D., Tompkins, P.: Model-unified planning and execution for distributed autonomous system control. In: *Workshop on Spacecraft Autonomy: Using AI to Expand Human Space Exploration*. AAAI Press (2006)
3. Browning, B., Bruce, J., Bowling, M., Veloso, M.: Stp: Skills, tactics and plays for multi-robot control in adversarial environments. *IEEE Journal of Control and Systems Engineering* 219, 33–52 (2005)
4. Baier, C., Katoen, J.P.: *Principles of Model Checking*. The MIT Press (2008)
5. Cesta, A., Cortellessa, G., Fratini, S., Oddi, A.: Developing an End-to-End Planning Application from a Timeline Representation Framework. In: *Proceedings of the 21st Innovative Applications of Artificial Intelligence Conference, IAAI 2009, Pasadena, CA, USA (2009)*
6. Cimatti, A., Roveri, M., Bertoli, P.: Conformant planning via symbolic model checking and heuristic search. *Artif. Intell.* 159(1-2), 127–206 (2004)
7. Firby, R.J.: Task networks for controlling continuous processes. In: *Artificial Intelligence Planning Systems: Proceedings of the First International Conference*, pp. 49–54. Morgan Kaufmann Pub. (1994)
8. Georgeff, M.P., Lansky, A.L.: Reactive reasoning and planning. In: *Proceedings of AAAI 1987 Sixth National Conference on Artificial Intelligence, Seattle, WA (USA)*, pp. 677–668 (July 1987)
9. Ghallab, M., Laruelle, H.: Representation and control in IxTeT, a temporal planner. In: *Proceedings of the 2nd International Conference on AI Planning Systems (1994)*
10. Guzman, C., Alcazar, V., Prior, D., Onaindia, E., Borrajo, D., Fernández-Olivares, J., Quintero, E.: Pelea: a domain-independent architecture for planning, execution and learning. In: *Scheduling and Planning Applications woRKshop (SPARK) ICAPS 2012 (2012)*
11. Sycara, K.P., Paolucci, M., Van Velsen, M., Giampapa, J.A.: The retsina mas infrastructure. *Autonomous Agents and Multi-Agent Systems* 7(1-2), 29–48 (2003)
12. Lesser, V., Decker, K., Wagner, T., Carver, N., Garvey, A., Horling, B., Neiman, D., Podorozhny, R., Prasad, M.N., Raja, A., Vincent, R., Xuan, P., Zhang, X.Q.: Evolution of the gpgp/taems domain-independent coordination framework. *Autonomous Agents and Multi-Agent Systems* 9(1-2), 87–143 (2004)
13. Sebastian Sardiña, L.P.: A bdi agent programming language with failure handling, declarative goals, and planning. *Autonomous Agents and Multi-Agent Systems* 23(1), 18–70 (2011)
14. Simmons, R.: Concurrent planning and execution for autonomous robots. In: *IEEE International Conference on Robotics and Automation*, pp. 46–50 (1992)
15. Such, J.M., Garcia-Fornes, A., Espinosa, A., Bellver, J.: Magentix2: a Privacy-enhancing Agent Platform. *Engineering Applications of Artificial Intelligence (2012)*
16. Wilkins, D.E., Myers, K.L.: *A multiagent planning architecture (1998)*

A Discussion on Trust Requirements for a Social Network of Eahoukers

Manuel Graña¹, J. David Nuñez-Gonzalez¹, and Bruno Apolloni²

¹ Grupo de Inteligencia Computacional (GIC), Universidad del País Vasco, Spain
manuel.grana@ehu.es

² Dept. of Computer Science, University of Milano, Milano, Italy

Abstract. A social network of eahoukers is intended to benefit from the socially generated knowledge to deal with the home appliances in a domestic environment. The entire system being developed in the SandS project has diverse facets, in this paper we focus on a discussion of the trust requirements from several points of view. Trust has been studied for a long time in different contexts. In this paper we review some definitions and ideas related to trust, as a basis for the desired discussion, as a first step previous to any implementation.

1 Introduction

When we make a decision about something of unknow consequences, we assume the risk due to the uncertainty about the results that we expect. If we have information sources to help us to predict the outcome of our decision, we will use them as long as we trust them. Trust is built in a feedback process, in which positive or negative results confirming the judgement given by the information sources will increase our trust in the information source. Conversely, if the results go against the expectations built from the information sources our trust will diminish. This paradigm is extensively studied in the construction of ad-hoc networks [9][10]. [3] organizes trust research in four major areas: (1) policy-based trust, (2) reputation based trust, (3) general models of trust and (4) trust information resources, related with the following applications: networking, semantic web, computational models, game theory and agents, software engineering and information resources. More specific application examples are education [1], Medical Sensor Networks [15], Industrial Digital Ecosystems[12], e-commerce. [17].

In the context of home appliance management and interaction in the domestic environment, the word “eahouker” has been coined in the project SandS meaning “easy house worker”, that is, a house worker that is enhanced by the help of software and social assistance. In the Social and Smart vision, eahoukers deal with their home appliances benefitting from the knowledge generated by their social interactions, empowered by an intelligent layer that produces new solutions when the responses given by the socially gathered database do not answer the question posed by the user. In this paper, we discuss some aspects of trust in this system, as a previous analysis step towards implementation.

2 Trust Related Definitions and Background

Traditionally, trust has been a subject of study for four different areas of knowledge: social psychology, philosophy, economics and market research [6], however it is gaining presence in technological domains, such as communications [10]. We skip here the mathematical definitions of trust [21][13], resorting to some intuitive informal definition, such as “the degree of subjective belief about the behaviors of (information from) a particular entity”[11], “the expectation that a service will be provided or a commitment will be fulfilled” [16].

2.1 Trust Properties

The three main trust properties that are relevant to algorithm development for treating with it are transitivity, asymmetry, and personalization”[14]. Additionally subjectivity, dynamicity, and context-dependency [10], reflexivity [2,13], non antisymmetry, time-based aging and distance-based aging [2] may be considered.

Transitivity In simplified form, mathematical transitivity means that if $A \rightarrow B$ and $B \rightarrow C$ then $A \rightarrow C$. The trust relation does not support transitivity, quoting [18]: “Alice may trust Bob about movies, but not trust him at all to recommend other people whose opinion about movies is worth considering or not trust other people that Bob recommended as much as she trusts Bob”. In fact, trust diminishes [18,22] as the chain of trust recommendations increases in some exponential law of the length of the trust path.

Asymmetry Trust does not have to be a symmetrical concept, in other words, two entities need not have the same degree of trust in each other. A typical example is that in a hierarchical environment the degree of trust between the supervisor and the employee is different [1,14].

Personalization-Subjectivity [14] and [10] use a different approaches to explain this property. On the one hand, for [14] trust is inherently a personal opinion. Two entities A and B could have a different opinion about the trustworthiness of another entity C. On the other hand, an entity A could trust another entity B with a certain degree of trust [10].

Dynamicity Trust should be expressed as a continuous variable, rather than as a binary or even discrete-valued entity. A continuous valued variable can represent uncertainly better than a binary variable. [1] [10].

Context-dependency An entity can trust other entity for some tasks but not for other tasks[5]. For example, a node A from a network can trust other node B to ask for authentication tasks but not for key management tasks.

Reflexivity considering internal actions, if agents trust themselves we have reflexivity of the trust relation [13], which may be stated as the fact that the trust value of A on itself for any context is 1 [2].

Non-antisymmetry “If A trust B and B trust A, that does not indicate that $A = B$.” [2].

Time-based aging [2]: “The trust value of A on B for a specific context C decreases with the passage of time”. The trust on a piece of information

obtained at t_i time will decrease with the passage of time because in t_{i+1} some event may change the value of the associated objects. New pieces of information must be more trustable than the older ones.

Distance-based aging [2]: “If node A collects trust values about B from other nodes in the network (recommendation), the trust values collected from closer nodes should be counted with more weight compared to the values collected from distant nodes.”

2.2 Metrics and Models

Trust propagation and computing models are essentially directed graphs [19] where nodes represent entities and edges trust relations labeled by some trust metric values. Trust management may be centralized (when a central trusted arbitrator gives trust evaluations of the partners), or decentralized (where users are responsible for the calculation of their own trust values for any target). It can also be distinguished between reactive computation, that calculate trust values when explicitly required, and proactive, which compute continuously the trust values of the peers, aiming to avoid delay in trust decisions. Trust computing must be resilient to attacks, which may consist in node attacks giving arbitrary opinions on a compromised node, or edge attacks inserting false edges in the network. Adding positive and negative evidence to the trust computation allowing for an accurate and flexible model. In communication networks, trust computing must be built at the routing and protocol levels, as the basis for all the upper layers.

, in his own words, “important issues that should be considered by designers of trust metrics”. Then, there is a part of an example in Ad Hoc Networks using the given taxonomy. The second part shows a selection of trust metrics proposed by [22].

Zhang’s [22] gives five types of metrics based on the quoted references. The metrics are: binary state metric, scaled metric, probability metric, hybrid or multi-metric trust and value of metric.

- Binary State Metric. This metric uses binary states 0 and 1 to express trust and distrust, which in some systems, is only considered as vote or observation. Therefore, binary state opinions are the building blocks of more abstract trust computing.
- Discrete Scale Metric. Allows to choose an option in a given range. Once the choice done, we can convert it to a quantity value, usually discrete.
- Probabilistic Metric usually represents the probability of a evaluated target participant performing actions as the evaluating participant expects.
- Hybrid or multi-metric trust This is to “use multiple metrics as a trust tuple to express more comprehensive trust”. For example, [19] uses trust and confidence to form an opinion space.
- Negative Values Negative trust value can be interpreted as distrust. “Some researches state the necessary of negative trust value to express bad impression.

However, introducing negative trust value also brings in vulnerability. Because generally negative of negative will produce positive result, malicious participants can employ this feature to manipulate trust values in order to promote their companies trust value”.

2.3 Models

There are many trust models in the literature. For example, [19,9,20,8,7] propose different trust models. We select two models.

Marsh’s Model. Proposed by Marsh in 1994 in his Phd, it is considered the first prominent, comprehensive, formal, computational model of trust [3,4]. It consists in a set of variables and ways to combine them arriving to a normalized value in the range $[-1, 1]$. Marsh identified three types of trust: basic, over all contexts; general, between two people and all their contexts occurring together; and situational, between two people in a specific context.

Bharadwaj’s Model. A fuzzy computational model for trust and reputation concepts is proposed in [4]. Quoting him, “The most appropriate property to define the symmetric part is the reciprocity while the partner’s experience defines the asymmetric part. The reciprocity is the mutual favor or revenge and therefore to model it, we need to find the agreement (both individuals are satisfied or unsatisfied) and disagreement (only one of them is unsatisfied) between two partners. To do so, we can define two fuzzy subsets on each partner’s ratings (universe of discourse), namely satisfied and unsatisfied. The membership values of satisfied and unsatisfied fuzzy subsets for a given encounter always sum up to one, for example, 70% satisfaction indicates 30% dissatisfaction. From these two fuzzy sets we can find the agreement and disagreement between the two partners”.

3 A Conceptual Map Description of a SandS Session

Conceptual maps are an useful tool to describe relations between concepts <http://cmap.ihmc.us/>. Here we use them to specify the flows of data and operations that constitute a working session of SandS, this use may be seen as an abuse of the tool by orthodox conceptualizers. We are not considering the initialization phase, when the user is engaging the Social Network, or when the databases of recipes and task descriptions are being initially feed either by appliance manufacturers or by enthusiastic eahoukers. Including this transitory phase in this specification would only introduce uninformative complications. Figure 1 contains the conceptual map graph. The main elements are highlighted in red: the eahouker and the appliance, in fact all SandS is designed to mediate between them. Green boxes contain explicitly active computational modules such as the natural language processing module, the task and recipe managers, the networked intelligence and the domestic middleware. The blue circle highlights

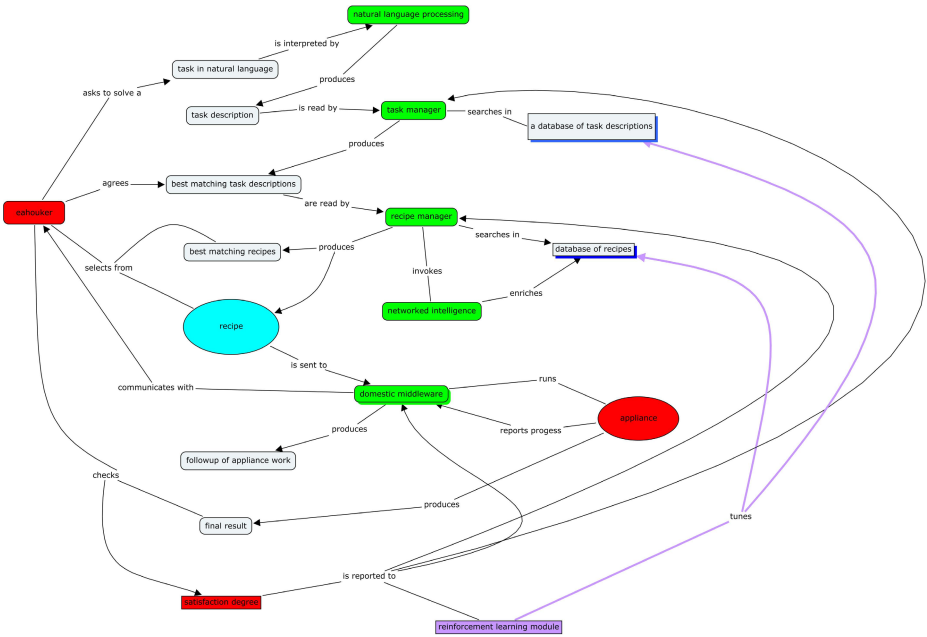


Fig. 1. Description of a SandS session by a conceptual map

the instrumental key of the system: the appliance recipe. The magenta box denotes a hidden reinforcement learning module which the eahouker is not aware of.

The SandS session is started by the eahouker stating a task in natural language. The natural language processing module analyzes this expression obtaining a task description which is suitable for formal search in databases. The task manager searches into a task database looking for the best match to the proposed task. If there is an exact match life will be easy for the recipe manager which needs only to retrieve the corresponding recipe. In general, the task manager will select a collection of best matching task descriptions which can be presented (or not) to the eahouker to assess the accuracy of the interpretation of his intentions by the system. The eahouker may agree to the best matching task descriptions. The recipe manager reads them and proceeds to search into the recipe database looking for best matches, or proceeds to request from the networked intelligence the enrichment of the recipe database with new solutions that may better fulfill the task posed by the user, or the best matching tasks. The recipe manager produces a selection of recipes and a best matching recipe. For the engaged user, the selection of recipes may allow her to reason about them and influence the recipe choice, it may even be an additional source of feedback to the networked intelligence.

When the recipe is selected, it is downloaded to the appliance via the domestic middleware, which controls its execution including the eventual communication with the user to operate the appliance (i.e. opening the appliance door).

The domestic middleware produces a monitoring followup of appliance function which may be shown to the user to keep her informed of progress, expected time to completion, etc. The appliance produces a final result, which is returned to the eahouker. Then the eahouker expresses her satisfaction, which is the main feedback for all processes reported to the domestic middleware, the recipe manager, the task manager and the reinforcement learning module. This last module uses this satisfaction reward value to fine tune the recipe and task databases to the user personality.

4 Discussion

The above session description allows to determine several critical interactions where trust can be introduced to modulate and control the responses of the system and the user. First, let us consider the construction of the tasks and recipes databases from eahouker interactions. There are two ways in which trust is relevant here. (a) the identification of rogue users that may try to sabotage competitors' appliances, (b) the quality of the recommendations coming from an specific users, and (c) the consensus between users, meaning that they agree on the quality evaluation of the results. The first becomes from external spurious sources, that may need the intervention of the networked intelligence as a filter to asses the likelihood that the entered information is from a rogue user. The second is relevant due to the amateur role of the eahoukers, they are peer users contributing information to a common pool, some of them may be unknowing and misleading. Repeated bad advices may be filtered by the social interaction by some reputation system (under some privacy constraints that the system must enforce to avoid unlawful situations). The third, is more subtil, because it refers to the personal tuning between eahoukers. One user may be giving good advice, but the criteria of other user may not agree on the quality of the results. This latter situation implies that trust values may act as modulators of the eahouker contributions.

The reinforcement module may play a role in the underlying computation of trust values according to the feedback satisfaction provided by the user. It will be feed sometimes with explicit rewards that must be propagated to the appropriate recipes in a backward fashion. This tuning amounts to a personalization of the responses of the system to the specific criteria of an eahouker.

The last item of trust corresponds to the networked intelligence. Users stablish their trust on the creative recipes that may be created by it, and these trust values may be used by the machine learning algorithms to perform adaptations of the rules and algorithms carrying the actual recipe generation. The bucket brigade problem that arises trying to perform such adaptation is among the most exciting challenges in the project. How to formulate trust and satisfaction, and how to extract them from the user interaction? Those are key questions being examining for whom we expect answers in the project lifetime.

References

1. Adams, W.J., Hadjichristofi, G.C., Davis IV, N.J.: Calculating a node's reputation in a mobile ad hoc network. In: 24th IEEE International Performance, Computing, and Communications Conference, IPCCC 2005, pp. 303–307 (April 2005)
2. Ahamed, S.I., Haque, M.M., Endadul Hoque, M., Rahman, F., Talukder, N.: Design, analysis, and deployment of omnipresent formal trust model (ftm) with trust bootstrapping for pervasive environments. *Journal of Systems and Software* 83(2), 253–270 (2010); *Computer Software and Applications*
3. Artz, D., Gil, Y.: A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web* 5(2), 58–71 (2007); *Software Engineering and the Semantic Web*
4. Bharadwaj, K.K., Al-Shamri, M.Y.H.: Fuzzy computational models for trust and reputation systems. *Electronic Commerce Research and Applications* 8(1), 37–47 (2009)
5. Bhargava, B., Lilien, L., Rosenthal, A., Winslett, M., Sloman, M., Dillon, T.S., Chang, E., Hussain, F.K., Nejdil, W., Olmedilla, D., Kashyap, V.: The pudding of trust (intelligent systems). *IEEE Intelligent Systems* 19(5), 74–88 (2004)
6. Blomqvist, K.: The many faces of trust. *Scandinavian Journal of Management* 13(3), 271–286 (1997)
7. Can, A.B., Bhargava, B.: Sort: A self-organizing trust model for peer-to-peer systems. *IEEE Transactions on Dependable and Secure Computing* 10(1), 14–27 (2013)
8. Chadwick, D.W., Young, A.J., Cicovic, N.K.: Merging and extending the pgp and pem trust models—the ice-tel trust model. *IEEE Network* 11(3), 16–24 (1997)
9. Chang, B.-J., Kuo, S.-L.: Markov chain trust model for trust-value analysis and key management in distributed multicast manets. *IEEE Transactions on Vehicular Technology* 58(4), 1846–1863 (2009)
10. Cho, J.-H., Swami, A., Chen, I.-R.: A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys Tutorials* 13(4), 562–583 (2011)
11. Cook, K.S. (ed.): *Trust in Society*, New York. Russell Sage Foundation Series on Trust, vol. 2 (February 2003)
12. Fachrunnisa, O., Hussain, F.K.: A methodology for maintaining trust in industrial digital ecosystems. *IEEE Transactions on Industrial Electronics* 60(3), 1042–1058 (2013)
13. Gai, X., Li, Y., Chen, Y., Shen, C.: Formal definitions for trust in trusted computing. In: 2010 7th International Conference on Ubiquitous Intelligence Computing and 7th International Conference on Autonomic Trusted Computing (UIC/ATC), pp. 305–310 (October 2010)
14. Golbeck, J.: Computing with trust: Definition, properties, and algorithms. In: *Securecomm and Workshops*, August 28–September 1, pp. 1–7 (2006)
15. He, D., Chen, C., Chan, S., Bu, J., Vasilakos, A.V.: A distributed trust evaluation model and its application scenarios for medical sensor networks. *IEEE Transactions on Information Technology in Biomedicine* 16(6), 1164–1175 (2012)
16. Hoffman, L.J., Lawson-Jenkins, K., Blum, J.: Trust beyond security: an expanded trust model. *Commun. ACM* 49(7), 94–101 (2006)
17. Manchala, D.W.: E-commerce trust metrics and models. *IEEE Internet Computing* 4(2), 36–44 (2000)
18. Sun, Y.L., Yu, W., Han, Z., Liu, K.J.R.: Information theoretic framework of trust modeling and evaluation for ad hoc networks. *IEEE Journal on Selected Areas in Communications* 24(2), 305–317 (2006)

19. Theodorakopoulos, G., Baras, J.S.: On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications* 24(2), 318–328 (2006)
20. Wang, X., Liu, L., Su, J.: Rlm: A general model for trust representation and aggregation. *IEEE Transactions on Services Computing* 5(1), 131–143 (2012)
21. Xiu, D., Liu, Z.: A formal definition for trust in distributed systems. In: Zhou, J., Lopez, J., Deng, R.H., Bao, F. (eds.) *ISC 2005*. LNCS, vol. 3650, pp. 482–489. Springer, Heidelberg (2005)
22. Zhang, P., Durrezi, A., Barolli, L.: Survey of trust management on various networks. In: *2011 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, June 30–July 2, pp. 219–226 (2011)

Querying on Fuzzy Surfaces with Vague Queries

Jan Caha and Jiří Dvorský

Department of Geoinformatics, Faculty of Science, Palacký University in Olomouc
17. listopadu 50, 771 46, Olomouc, Czech Republic
{jan.caha,jiri.dvorsky}@upol.cz

Abstract. The aim of the study is to present utilization of Possibility theory to evaluate soft spatial queries on Fuzzy Surfaces. Fuzzy Surfaces are constructed from incomplete datasets or from data that contain uncertainty that is not of statistical nature. Soft spatial queries are common in geography because a lot of classes that should be found in the data have naturally vague definitions or are defined by expert opinion in term of interval rather than exact threshold. Soft thresholds and Surfaces with uncertainty can be expressed with use of Fuzzy Numbers. To evaluate their exceedance or ranking the procedures from Possibility theory are utilized. The whole concept is shown on a Case study.

Keywords: fuzzy surface, soft queries, uncertainty, possibility theory.

1 Introduction

In geoinformatics it is often case that the information used for modelling is incomplete. For example points used to the model surface are only samples from the set of all points on the surface and their precision can be unknown. From such inputs it is not possible to create precise and absolutely certain surface. It is necessary to include the uncertainty, that has origin in both input data and interpolation process, in the surface. Since none of the mentioned types of uncertainty is of statistical nature it is convenient to conceptualize the uncertainty by an alternative method i.e. fuzzy set theory.

Spatial querying frequently leads to various categorizations of the result into sets with meaning - appropriate, more appropriate, rather inappropriate, inappropriate, etc. In such case rather than creating several sets it is much more practical to allow gradual transition from inappropriate to appropriate results. This can be done using so called soft thresholds. Those allow also modelling of vague terms like “steep slopes”.

In order to allow spatial queries with soft thresholds on *Fuzzy Surfaces* it is necessary to have complex system that allows ranking of fuzzy sets. Several such systems exists but the most complex and appropriate is the one using *Theory of possibility*. This allows the logically correct and complete evaluation of such spatial queries.

The structure of article is following: Sections 2 and 3 offers brief summary of Fuzzy Sets, Fuzzy Numbers, Possibility Theory and Pairwise comparison of

Fuzzy Numbers. Section 4 summarizes information about vagueness in geography and the need of soft queries for spatial decision support. Case study on the matter is shown in section 5 and the results are discussed and evaluated in sections 6 and 7.

2 Fuzzy Sets

Fuzzy set is a special case of set that does not have strictly defined criterion of membership. For example set of “large numbers” or “steep slopes” do not have strict threshold but rather a transitional interval where objects have increasing or decreasing value of membership. Fuzzy set is determined by the membership function, which is defined as mapping

$$\mu_{\tilde{A}} : U \longrightarrow [0, 1] \tag{1}$$

that indicates that element x of universe U has membership value $\mu_{\tilde{A}}(x)$ from the interval $[0, 1]$ [19]. Elements with $\mu_{\tilde{A}}(x) = 0$ do not belong to the set \tilde{A} , while elements $\mu_{\tilde{A}}(x) = 1$ completely belong to the set. Other membership values indicates partial membership in the set. Fuzzy set can be expressed as pairs of elements and their membership values

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in U, \mu_{\tilde{A}}(x) \in [0, 1]\}. \tag{2}$$

or by exact the definition of membership function [12].

2.1 Fuzzy Numbers

Fuzzy number is a special case of fuzzy set that represents vague, imprecise or ill-known value[5]. In order to be a fuzzy number the fuzzy set has to satisfy several conditions. The universe on which the set is defined should be real numbers \mathbb{R} . Its height according to

$$\text{hgt}(\tilde{A}) = \sup \{\mu_{\tilde{A}}(x) \mid x \in U\} \tag{3}$$

must be equal to 1. So that there is at least one x with full membership to the set \tilde{A} . The fuzzy set also has to be convex, which it is, if the condition

$$\mu_{\tilde{A}}(\lambda x_1 + (1 - \lambda)x_2) \geq \min((\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)) \quad \forall \lambda \in [0, 1], \forall x_1, x_2 \in U \tag{4}$$

is satisfied. The last condition is that the membership function $\mu_{\tilde{A}}(x), x \in \mathbb{R}$ must be at least piecewise continuous. If the fuzzy set satisfy all those conditions it can be treated as a fuzzy number, which means that it can be used for further calculations [12].

3 Ranking of Fuzzy Numbers

To allow decision making based on fuzzy numbers there is a need for a system that will allow *ranking of fuzzy numbers*. There are several such systems however most of them consider only one point of view on the problem [5]. The complete set of ranking indices in the framework of possibility theory was proposed in [5]. This ranking system is based on the possibility theory. First it is needed to explain basic principles of theory of possibility.

3.1 Possibility and Necessity Measures

Let \tilde{F} be a normalized fuzzy subset of universe U which is characterized by the membership function $\mu_{\tilde{F}}$. This fuzzy sets act as a fuzzy restriction [5]. Let X be a variable from U . Then the *possibility measure* derived from the membership function $\mu_{\tilde{F}}$ by

$$\Pi_{\tilde{F}}(X) = \sup_{x \in X} \mu_{\tilde{F}}(x) \quad \forall X \subseteq U \tag{5}$$

and $\mu_{\tilde{F}}$ is a possibility distribution underlying $\Pi_{\tilde{F}}$ that can be denoted as $\pi_{\tilde{F}}$ [6]. If X is a fuzzy set \tilde{X} then this equation can be extended to

$$\Pi_{\tilde{F}}(\tilde{X}) = \sup_x \min(\mu_{\tilde{F}}(x), \mu_{\tilde{X}}(x)). \tag{6}$$

Let \overline{X} be complement of X and Π a possibility measure then set function \mathcal{N} defined by

$$\mathcal{N}(X) = 1 - \Pi(\overline{X}) \quad \forall X \subseteq U \tag{7}$$

is a *necessity measure* [6]. Such necessity measure can be also called certainty measure. If both X and F are fuzzy set \tilde{X}, \tilde{F} then this equation is extended to

$$\mathcal{N}_{\tilde{F}}(\tilde{X}) = 1 - \sup_x \min(\mu_{\tilde{F}}(x), 1 - \mu_{\tilde{X}}(x)). \tag{8}$$

3.2 Pairwise Comparison of Fuzzy Numbers

In order to asses the relative position of the fuzzy number \tilde{Y} to the fuzzy number \tilde{X} four indices are needed [5]. These are $\Pi_{\tilde{X}}([\tilde{Y}, \infty))$ and $\mathcal{N}_{\tilde{X}}([\tilde{Y}, \infty))$ to asses possibility and necessity that \tilde{X} is greater or at least equal to \tilde{Y} . And $\Pi_{\tilde{X}}([\tilde{Y}, \infty))$, $\mathcal{N}_{\tilde{X}}([\tilde{Y}, \infty))$ for evaluating strict exceedance of \tilde{Y} by \tilde{X} . Expressions for calculating those indices are following:

$$\Pi_{\tilde{X}}([\tilde{Y}, \infty)) = \sup_x \min(\mu_{\tilde{X}}(x), \sup_{y \leq x} \mu_{\tilde{Y}}(y)) \tag{9}$$

$$\mathcal{N}_{\tilde{X}}([\tilde{Y}, \infty)) = \inf_x \max(1 - \mu_{\tilde{X}}(x), \sup_{y \leq x} \mu_{\tilde{Y}}(y)) \tag{10}$$

$$\Pi_{\tilde{X}}(\tilde{Y}, \infty)) = \sup_x \min(\mu_{\tilde{X}}(x), \inf_{y \geq x} 1 - \mu_{\tilde{Y}}(y)) \tag{11}$$

$$\mathcal{N}_{\tilde{X}}(\tilde{Y}, \infty) = \inf_x \max(1 - \mu_{\tilde{X}}(x), \inf_{y \geq x} 1 - \mu_{\tilde{Y}}(y)). \tag{12}$$

These results answer the questions “is \tilde{X} greater than \tilde{Y} ” and “is \tilde{X} strictly greater than \tilde{Y} ” in terms of both possibility and necessity (Fig. 1). Details on the implementation, proofs and process of answering inverse problem are provided in [5] and [6].

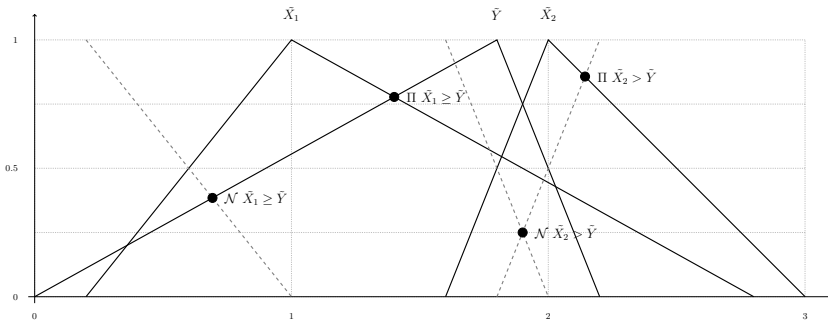


Fig. 1. Four indices comparing \tilde{X}_1 and \tilde{X}_2 to \tilde{Y}

4 Vagueness in Geography

Vagueness is widely discussed topic in geography with applications to modelling fuzzy geographical regions [8,10,11], surfaces with uncertainty [16,17] and decision making based on those vague datasets [7,18]. All those possible applications are essential for geography, because within it is domain there are many concepts that are naturally vague [8].

GIS (Geographical Information Systems) are valued tools for supporting *decision making* with spatial data however the reliability of results should be questioned [13]. All the data and process are usually treated as certain and absolutely correct, event through that none of them actually are either completely certain of correct [8,11,13]. Besides the question of modelling data and process with *uncertainty* and/or *vagueness* there is also the question of querying the results. Commonly used concepts do not allow almost any flexibility when querying the data. However the flexibility of queries is a necessary concept to access more complex decision support [2]. So far there are several studies dealing with quoting fuzzy geographical data [1,8,18] but there is no mention of using *Possibility theory* and its measures of possibility and necessity. Use of Possibility theory allows modelling of vague data and also vague queries [7], thus it seems as a reasonable tool, that should be used while quoting uncertain datasets with vague queries.

4.1 Fuzzy Surfaces

In GIS many variables are modelled as surfaces using so called *field model*. Field model is partitioning of a geographical space into finite number of spatial entities, usually squares [14]. Surface itself is represented by the mathematical function $Z = f(x, y)$. Such function defined on a geographical space assigns each of this spatial entities (*cells*) value Z . In case of fuzzy surfaces the resulting surface incorporates uncertainty of input data and/or processes of interpolation and the result value of each cell of the grid is a fuzzy number \tilde{Z} . Such surfaces can be further analysed by means of *fuzzy arithmetic* to derive slope, visibility and other parameters [3,16,17].

4.2 Soft Spatial Queries

Usual *spatial queries* aim to select areas that meet one or more conditions. The condition is usually defined as a value being higher or smaller than given *threshold*. Such queries cannot quite well introduce any measure of preference in the result, because they are based on classical logical expressions [2,7]. But many decision making situations that involve spatial data are not well suited for such crisp queries, because the crisp query might be far to restrictive for such utilization [18].

Classic query often looks like “is variable X higher (or lower) then threshold Y ?”. In such case it does not matter what is the difference between X a Y , if the X is smaller then it is rejected from the set. However there is a clear difference between $X_1 = 3.14$ and $X_2 = 5.999$ when they are compared to the threshold $Y = 6$. While X_1 is clearly smaller and should not be included in the set of numbers equal or higher than Y , X_2 is quite another matter. Indeed it is lower then Y but the difference is so small that X_2 is almost indistinguishable from the threshold. For a complex decision making processes it would be much better to specify the threshold as a fuzzy number [7]. In such case a triangular fuzzy number with support values [5, 7] and kernel value 6 can be used as an approximation to the original threshold. Evaluating if the specific value of X is lower (or higher) then such threshold is then matter of comparing real value and Fuzzy number and it can be done in terms of possibility and necessity [5].

The problem is even more complex if the input is a fuzzy number. Then both the value and the threshold are fuzzy numbers \tilde{X}, \tilde{Y} . The comparison $\tilde{X} > \tilde{Y}$ then can be made according to [5] by four indices: *possibility*, *necessity*, *strict possibility* and *strict necessity* of the exceedance of \tilde{Y} by \tilde{X} .

The reasons for creating fuzzy threshold can be summarized as following:

- the concept of Y is naturally fuzzy i.e. definition of “steep slopes”,
- there are more than one acceptable definitions of Y and there is no indications that any of them is more correct or precise than the others,
- Y is based on the expert opinion that is provided as an interval of values rather then precise value, or there is a need to merge definitions of Y from several such expert opinions.

These findings are supported by numerous studies [2,7,11].

Suppose that we have a fuzzy surface represented by a field model that has N rows and M columns, with each cell denoted by $\tilde{C}_{i,j}$ where $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, M$. In order to perform a soft spatial query, there is a need to compare all the $N \times M$ fuzzy numbers to the given soft threshold \tilde{T} in terms of Eqs. (9), (10) in case $\tilde{C}_{i,j} \geq \tilde{T}$ to figure out possibility and necessity of exceedance. And in terms of Eqs. (11), (12) in case $\tilde{C}_{i,j} > \tilde{T}$, which denotes possibility and necessity of the strict exceedance of \tilde{T} by $\tilde{C}_{i,j}$.

5 Case Study

The aim of the case study is to show how soft queries can be used to query fuzzy surfaces. The task will be to find out areas with higher than “medium slope” on a fuzzy surface (all the slope values are presented in degrees). Such query can be for example part of a complex decision of finding suitable areas for waste disposal site. Such object cannot be located on “steep slopes” so all areas that have slope higher then “medium” have to be eliminated from the set of possible locations. Since there is no universal definition of “medium slope” it will be defined according to the expert opinion as a triangular fuzzy numbers with support values $[5^\circ, 9^\circ]$ and kernel value 7° . This definition states that values below 5° clearly are not “medium slope”, value 7° is what can be considered as crisp threshold and values higher then 9° are clearly above the definition of “medium slope”. This is a natural definition of the vague object or objects with fuzzy borders according to [19].

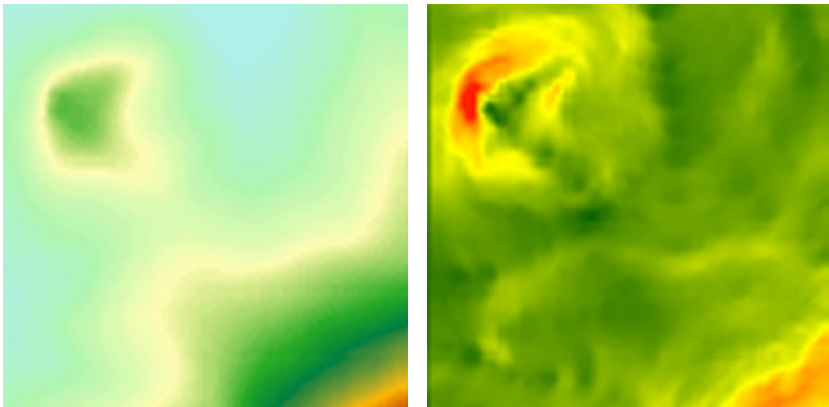


Fig. 2. Visualization of kernel values of the fuzzy surface (left) and fuzzy slope (right). Terrain colours are from light blue (low values) to brown (high values), while slope is from green (low) to red (high).

For the purpose of this case study the fuzzy surface was build using the process described in [4]. The kernel values of fuzzy numbers are interpolated by kriging from the input data and the support values are constructed from the kriging standard deviation supplemented with an expert knowledge. The triangular fuzzy number for each cell is constructed according to the formula:

$$\tilde{C}_{i,j} = [C_{i,j} - (E_d + \varepsilon_{i,j} * E_e), C_{i,j}, C_{i,j} + (E_d + \varepsilon_{i,j} * E_e)] \quad (13)$$

where E_d is a minimal estimation error that is assumed, E_e is an error that arise from the uncertainty of prediction. $\varepsilon_{i,j}$ denotes the normalized standard error of kriging at cell i, j . E_d value was defined as 0.5 meters, E_e as 0.75 meters. These values are based on the expert opinion [4]. The extent of support of the fuzzy number at the worst case ($\varepsilon_{i,j} = 1$) is equal to the half of the contour lines interval, that follows the methodology of [11] and also [15]. From this fuzzy surface the fuzzy slope was calculated. The details of the use of fuzzy arithmetic for calculating slope of fuzzy surfaces are provided in [3] and [17]. Kernel values of both surface and slope are shown on Fig. 2.

Now the comparison of the data and the soft threshold can be made. First calculations according to Eqs. (9) and (10) are done (Fig. 3). Possibility of exceedance shows areas that have at least some chance of fulfilling the condition however if their necessity is zero then it is not strongly supported. If the necessity is higher then 0 then there are stronger indicators supporting the condition. The value of necessity 1 gives quite strong evidence of exceedance.

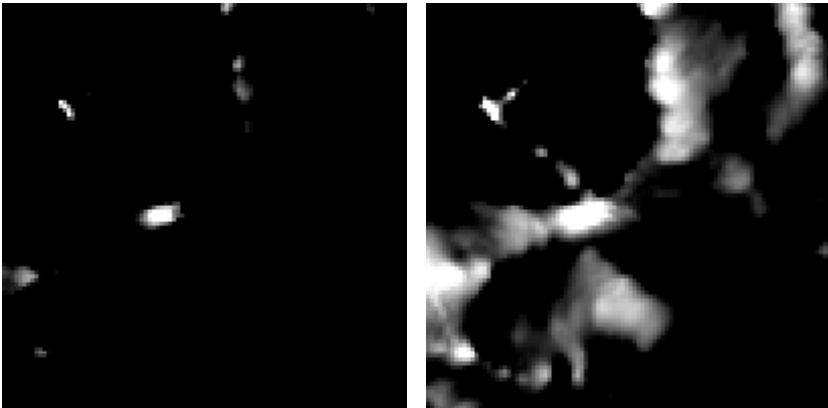


Fig. 3. Visualization of possibility (left) and necessity (right) of exceedance of the threshold. Black colour means value 1 and white 0

If strict exceedance of the threshold by the values is required then Eqs. (12) and (11) are used (Fig. 4). With those indicators even the possibility of strict exceedance higher then 0 is quite strong support of the fulfilment of the condition. And necessity value of 1 indicates the absolute fulfilment of the condition. Visualization of comparison of one cell value to the threshold can be seen on Fig. 5.

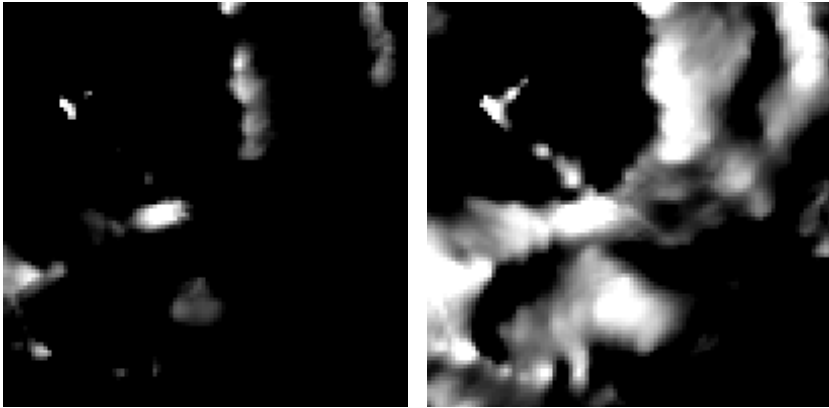


Fig. 4. Visualization of possibility (left) and necessity (right) of strict exceeder of the threshold. Black colour means value 1 and white 0

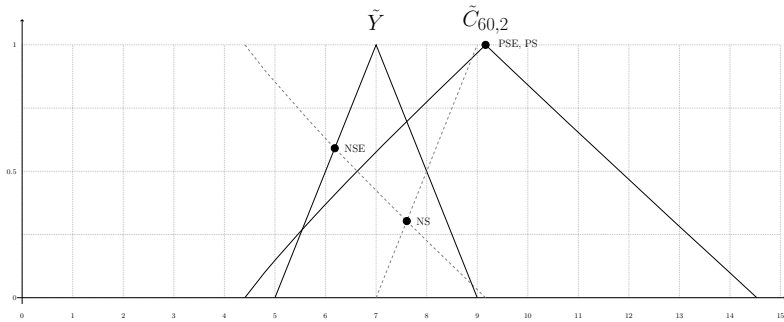


Fig. 5. Four indices comparing one cell of the grid $\tilde{C}_{60,2}$ to the threshold \tilde{T}

6 Discussion

In geoinformatical sciences there are two main approaches to the problem of *uncertainty propagation*. These are Monte Carlo simulation (based on statistics) and Analytical approach [15]. Monte Carlo method is often used because of its simplicity in terms of implementation while Analytical approach is quite complex and therefore used rather rarely. Besides those there is a new trend of modelling uncertainty with use of fuzzy set [11,16]. However use of *fuzzy mathematics* for uncertainty propagation is not very common for geographical problems [9], but in other disciplines these methods were proved to be useful [12]. If these new methods are used for experimental uncertainty propagation, and according to Heuvelink [13] there is a need for new approaches to this problem, then new methods of evaluating and querying the results are necessary.

The proposed approach of using soft queries on fuzzy surfaces is distantly related to the process of evaluating of cumulative distribution function (CDF)

of statistical data. If the uncertainty is propagated through the operation using Monte Carlo and user wants to know how probable is exceedance of specific threshold then the answer to this question is obtained from the CDF of result of experiment. In the case of fuzzy sets the analogy is found within the *Possibility theory* in measures of possibility and necessity. This approach also allows the comparison of fuzzy numbers, which is very useful for creating soft thresholds. This approach offers more information for the decision maker when the uncertainty in the input data is present and the decision criterion cannot be specified exactly or such specific definition would be too restrictive.

7 Conclusion

The proposed approach allows answering *vague spatial queries on Fuzzy Surfaces*. Four indices are used to reach this objective, each of this indices takes value from the interval $[0, 1]$ where 1 means the complete fulfilment of the rule while 0 means failure to reach the rule. These four indices form two pairs of possibility and necessity measures that complete each other. In the case of possibility value 1 and necessity value lower than 1 (such as in case shown on Fig. 5) there are some indicators that value $\tilde{C}_{60,2}$ can possibly be higher than the threshold but it is not necessary because both values of necessity of exceedance and strict necessity of exceedance are lower than 1 (0.59, respectively 0.30). This information is very useful because it tells the decision maker that it might be that the cell has slope higher than the threshold but that it is not absolutely sure, that it is so. Such areas can be subject to further examination, that will offer more information that will eliminate the uncertainty and allow the final decision to take place.

These four indices form natural ordering where the possibility of exceedance is the less restrictive, followed by the possibility of strict exceedance, then the necessity of exceedance and the necessity of strict exceedance is the most restrictive. If the Eq. (12) equals 1 it means that the soft threshold is exceeded certainly while value of Eq. (9) that equals 0 means that by no chance is the value higher than the threshold.

Proposed approach extends possibilities of *spatial decision support* by the use of mathematical methods for handling *uncertainty* and *vagueness* that is very common in *geographical sciences*. Further studies should be focused on more complex decisions than the simple example shown in the case study.

Acknowledgement. The authors gratefully acknowledge the support by the Operational Program Education for Competitiveness - European Social Fund (projects CZ.1.07/2.3.00/20.0170 and CZ.1.07/2.2.00/28.0078 of the Ministry of Education, Youth and Sports of the Czech Republic).

References

1. Boroushaki, S., Malczewski, J.: Using the fuzzy majority approach for GIS-based multicriteria group decision-making. *Computers & Geosciences* 36(3), 302–312 (2010)

2. Bosc, P., Kraft, D., Petry, F.: Fuzzy sets in database and information systems: Status and opportunities. *Fuzzy Sets and Systems* 156(3), 418–426 (2005)
3. Caha, J., Tuček, P., Vondráková, A., Paclíková, L.: Slope Analysis of Fuzzy Surfaces. *Transactions in GIS* 16(5), 649–661 (2012)
4. Caha, J., Tuček, P., Vondráková, A., Paclíková, L.: Fuzzy surface models based on kriging outputs. In: Růžička, J. (ed.) *Symposium GIS Ostrava 2012 - Proceedings Surface Models for Geosciences*, pp. 25–36. VSB - Technical University of Ostrava, Ostrava (2012)
5. Dubois, D., Prade, H.: Ranking Fuzzy Numbers in the Setting of Possibility Theory. *Information Sciences* 30(3), 183–224 (1983)
6. Dubois, D., Prade, H.: *Possibility Theory An approach to Computerized Processing of Uncertainty*. Plenum Press, New York (1986)
7. Dubois, D.: The role of fuzzy sets in decision sciences: Old techniques and new directions. *Fuzzy Sets and Systems* 184(1), 3–28 (2011)
8. Fisher, P.: Sorites paradox and vague geographies. *Fuzzy Sets and Systems* 113(1), 7–18 (2000)
9. Fisher, P.F., Tate, N.J.: Causes and consequences of error in digital elevation models. *Progress in Physical Geography* 30(4), 467–489 (2006)
10. Fisher, P., Cheng, T., Wood, J.: Higher order vagueness in geographical information: Empirical geographical population of type n fuzzy sets. *Geoinformatica* 11(3), 311–330 (2007)
11. Fonte, C.C., Lodwick, W.A.: Modelling the Fuzzy Spatial Extent of Geographical Entities. In: Petry, F., Robinson, V.B., Cobb, M.A. (eds.) *Fuzzy Modeling with Spatial Information for Geographic Problems*, pp. 120–142. Springer, Berlin (2005)
12. Hanss, M.: *Applied fuzzy arithmetic: an introduction with engineering applications*. Springer, Berlin (2005)
13. Heuvelink, G.B.M.: Analysing Uncertainty Propagation in GIS: Why is it not that Simple? In: Foody, G.M., Atkinson, P.M. (eds.) *Uncertainty in Remote Sensing and GIS*, pp. 155–165. Wiley, Chichester (2002)
14. Janoška, Z., Dvorský, J.: P systems: State of the art with respect to representation of geographical space. In: *CEUR Workshop Proceedings - 12th Annual Workshop on Databases, Texts, Specifications and Objects, DATESO 2012*, pp. 13–24 (2012)
15. Oksanen, J., Sarjakoski, T.: Error propagation of DEM-based surface derivatives. *Computers & Geosciences* 31(8), 1015–1027 (2005)
16. Santos, J., Lodwick, W.A., Neumaier, A.: A New Approach to Incorporate Uncertainty in Terrain Modeling. In: Egenhofer, M.J., Mark, D.M. (eds.) *GIScience 2002. LNCS, vol. 2478*, pp. 291–299. Springer, Heidelberg (2002)
17. Waelder, O.: An application of the fuzzy theory in surface interpolation and surface deformation analysis. *Fuzzy Sets and Systems* 158(14), 1535–1545 (2007)
18. Witlox, F., Derudder, B.: Spatial Decision-Making Using Fuzzy Decision Tables: Theory, Application and Limitations. In: Petry, F., Robinson, V.B., Cobb, M.A. (eds.) *Fuzzy Modeling with Spatial Information for Geographic Problems*, pp. 120–142. Springer, Berlin (2005)
19. Zadeh, L.A.: Fuzzy Sets. *Information and Control* 8(3), 338–353 (1965)

Best Fuzzy Partitions to Build Interpretable DSSs for Classification in Medicine

Marco Pota, Massimo Esposito, and Giuseppe De Pietro

Institute for High Performance Computing and Networking, ICAR-CNR
Via P. Castellino 111, Naples 80131, Italy

{marco.pota,massimo.esposito,giuseppe.depietro}@na.icar.cnr.it

Abstract. Decision Support Systems (DSSs) based on fuzzy logic have gained increasing importance to help clinical decisions, since they rely on a transparent and interpretable rule base. On the other hand, probabilistic models are undoubtedly the most effective way to reach high performances. In order to join positive features of both these two approaches, this work proposes a hybrid approach, consisting in transforming the functions describing posterior probabilities, into a combination of orthogonal fuzzy sets approximating them. The resulting fuzzy partition has double hopefulness: since it approximates posterior probabilities, it is able to model information extracted from a dataset in such a form that they can be used to run predictions, and since it is a set of normal, orthogonal and convex fuzzy sets, it can be interpreted as the set of terms of a linguistic variable. As a proof of concept, the method has been applied to a real-life application pertaining the classification of Multiple Sclerosis Lesions. The results show that this method is able to construct, for each one of the variables influencing the classification, interpretable *if-then* rules, with classification power comparable to that of a classical Bayesian model.

Keywords: probability, statistical learning, fuzzy partition, classification, linguistic variable, clinical DSS.

1 Introduction

Data-driven Decision Support Systems (DSSs) are gaining increasing importance in recent research [1]. In fields like medicine, DSSs based on fuzzy logic [2] are required, since they allow to operate with data affected by uncertainty and vagueness; moreover, they are characterized by a transparent and comprehensible knowledge base [3], showing a clear and logic justification for each conclusion, even to non-technical users. The knowledge base consists in the partition of the input variables into a collection of fuzzy sets, and rules correlating these fuzzy sets to classes, defined according to the *if-then* structure. Different approaches can be used to perform the fuzzy partitioning of numerical variables and to construct the fuzzy rule base [4-11]. However, this approach suffer of the drawback that interpretable membership functions are constrained (in particular, they should be normal, orthogonal, and convex [5]), thus the real trend of data is hard to be approximated. On the other hand,

statistical methods, which are mainly based on a Bayesian perspective [12], aim to obtain good performances. However, if a fuzzy knowledge base is constructed starting from statistical information [13-16], a good fitting of data is reached without fixing constraints, but fuzzy partitions are found which are not able to model interpretable terms of a linguistic variable. Another drawback of existing approaches is that they choose the most probable class as a result of the classifier, by comparing respective probabilities without calculating them, while in the medical field the physician could be not only interested in knowing the most probable class the patient belongs to, but also all the other possible classes should be always considered. Currently, at the best of our knowledge, very low effort was paid to the aim of merging characteristics of highly interpretable fuzzy models and highly performing statistical methods.

This work proposes a hybrid approach, to build transparent fuzzy DSSs for classification in medicine. The aim is to obtain a good classifier, with performances similar to those of a bayesian one, governed by a fuzzy knowledge base modeled by simple rules and interpretable fuzzy partitions, as required in medical field. Moreover, this approach allows not only to choose the most probable class, but also to calculate the probability of each class, given an incoming data item, therefore, its strength relies also on this kind of results, which is significantly attractive in the medical field. These peculiarities are achieved by performing the knowledge extraction in two steps. Firstly, a statistical approach is used to obtain a model of the probability of each class, given a generic point of the feature space. Then, such a mapping is correlated to constrained (normal, orthogonal, convex) fuzzy sets.

The rest of the paper is organized as follows. Section 2 describes classical fuzzy and statistical classifiers. In Section 3, the proposed approach is explained, while its application to Multiple Sclerosis Lesions (MSL) dataset is shown in Section 4. Finally, Section 5 concludes the work.

2 Background

In fuzzy systems, a transparent knowledge base is ensured [3]. Moreover, depending on their use, inference procedures are usually devoted to reach mainly one of the following objectives: *i*) the interpretability of the knowledge base, which can be obtained by using constrained fuzzy systems; *ii*) the “goodness” of the classifier (e.g., high classification rate (CR) [17], low squared classification error (SCE) [18]), which is undoubtedly optimized by using statistical approaches. As a consequence, the best approach depends on the objectives of the DSS.

2.1 Constrained Fuzzy Systems

The fuzzy knowledge is constituted by a model which allows to perform the decision-making process based on fuzzy logic. The knowledge extraction consists in different aspects, which can be performed in different ways [4-11]. Some basic information about the construction of interpretable fuzzy partitions and the rule-based fuzzy inference are summarized as follows.

Fuzzy partitioning consists in the construction of membership functions of fuzzy sets to be used in the rules. A fuzzy partition is considered as interpretable if its fuzzy sets model terms of a linguistic variable [5]. Fuzzy sets which model linguistic terms should satisfy, for each feature, the properties of normality (1) and orthogonality (2):

$$\forall t \in \{1, \dots, T\}, \begin{cases} \min\{\mu_t(x), x \in U\} = 0 \\ \max\{\mu_t(x), x \in U\} = 1 \end{cases}, \quad (1)$$

$$\forall x \in U, \sum_{t=1}^T \mu_t(x) = 1, \quad (2)$$

where x is the value of a variable describing the data item, U is the universe of discourse for that feature, $\mu_t(x)$ is the value of the membership function of the t -th fuzzy set, and T is the number of linguistic terms. Moreover, all fuzzy sets should be convex. Finally, the number of linguistic terms should be not too large, and a maximum of 9 terms is usually taken.

Membership functions $\mu_t(x)$ are typically represented by parametric functions, which could have different shapes (triangular, trapezoidal, bell-shaped and so on).

A simple fuzzy classifier comprises a set of fuzzy rules. In each rule, the antecedent defines the region of the H -dimensional feature space, while the consequent is a class label belonging to the set $\{c_1, \dots, c_C\}$ of the C classes. Here, simple rules made of only one antecedent are considered:

$$r_i(w_i): \text{If } x^{(h)} \text{ is } F_i^{(h)} \text{ then } y = c_k, \quad (3)$$

where $x^{(h)}$ is one of the components of the vector \mathbf{x} of variable values describing the data item to be classified, and $F_i^{(h)}$ is one of the antecedent fuzzy sets. A weight w_i can be associated to each rule in order to model a different impact, which can be defined by an expert or calculated using an optimization process. Note that *and* (*or*) connectives may be used to put more antecedents in the same rule; this can be modeled by applying a T-norm (S-norm) to the set of membership grades $\mu^{(h)}$ of $x^{(h)}$ to $F^{(h)}$. In the simple case of only one variable, the degree of activation of the i -th rule is:

$$\alpha_i(\mathbf{x}) = w_i \cdot \mu_i^{(h)}(x^{(h)}). \quad (4)$$

If different rules have the same class as consequent, an S-norm is applied to the activations of these rules to obtain the aggregated activation A_k of the k -th class:

$$A_k(\mathbf{x}) = \text{S-norm}[\delta_{1,k}\alpha_1(\mathbf{x}), \dots, \delta_{R,k}\alpha_R(\mathbf{x})], \quad (5)$$

where R is the number of rules and $\delta_{i,k}$ is equal to one if the consequent of the i -th rule is the class c_k , zero otherwise. Thus, consequent classes are activated with different grades. A single output of the fuzzy classifier can be determined by the “winner takes all” strategy, i.e. the output is the class that gets the highest degree of activation.

2.2 Fuzzy Systems Based on Statistical Inference

Fuzzy knowledge extraction methods based on statistical approaches [13-16] are particularly useful to optimize the performances of the system, since they are mainly based on Bayes' rule [12]:

$$\mathbf{x} \text{ is assigned to } c_k \Leftrightarrow p(c_k | \mathbf{x}) \geq p(c_i | \mathbf{x}) \quad \forall i \neq k . \quad (6)$$

In particular, a very simple form is reached by assuming independence of input variables, thus obtaining:

$$\mathbf{x} \text{ is assigned to } c_k \Leftrightarrow P(c_k) \prod_h p(x^{(h)} | c_k) \geq P(c_i) \prod_h p(x^{(h)} | c_i) \quad \forall i \neq k . \quad (7)$$

Naive Bayes classifier transforms (6) into (7), in order to obtain a compact formalization. Indeed, class probability distributions $p(x|c_k)$ are often assumed to have fixed shape, defined by parametric probability density functions (e.g. Gaussian PDFs), while posterior probabilities $p(c_k|x)$ are less appropriate to be modeled by imposing such an assumption. Together with this simplification, one is only able to choose the most probable among classes, while in fields like medicine, the user should consider all possible classes, which should be given as a result together with respective probability. Therefore, one should calculate an approximate, parametric or non-parametric, mapping of $p(c_i|x)$ as a function of x .

Fuzzy partitions for fuzzy inference can be constructed by using either class probability distributions $p(x|c_k)$ [13-15] or posterior probabilities $p(c_k|x)$ [16], but both could result to be not fully interpretable.

3 The Proposed Approach

In this work, a hybrid of fuzzy logic and statistical approach is presented, in order to construct a DSS particularly suitable for medical applications. It allows to extract, a knowledge base from a training dataset \mathbf{Z} made of data items $\mathbf{z}_j=[\mathbf{x}_j, y_j]$, with $\mathbf{x}_j=[x_j^{(1)}, \dots, x_j^{(H)}]$ and $y_j \in \{c_1, \dots, c_C\}$, $j=1, \dots, N$, where H is the number of data features and N is the number of data items. The extracted knowledge is made of interpretable fuzzy sets satisfying (1) and (2), and fuzzy rules, and it allows calculating probabilities that an incoming data item $\mathbf{x}=[x^{(1)}, \dots, x^{(H)}]$ belongs to classes $\{c_1, \dots, c_C\}$. In particular, the approach is described here for extracting one-dimensional models, therefore, the symbol x will substitute both $x^{(h)}$ and \mathbf{x} .

The proposed approach consists of three steps. Firstly, the functions describing posterior probabilities of classes, $p(c_k|x)$, as a function of the input feature, are calculated starting from the training dataset. Then, these functions are approximated with a combination of interpretable fuzzy sets. Finally, a weighted rule base is obtained, which allows to make inference about new incoming data items in terms of probabilities of different classes.

3.1 Functions Describing Posterior Probabilities

Posterior probabilities $p(c_k|x)$ are obtained here in a point-wise manner. In order to calculate them, the following procedure is adopted.

1. An equally-spaced grid of a number n of points x_ν is used, with $\nu=1,\dots,n$, representing the whole universe of discourse U . Therefore, each point is representative of an interval whose measure is M/n , where M is the measure of the whole U (it is always finite, because the assumption is made of having finite data items).
2. Kernel functions [19] are used to estimate the non-parametric function of class probability distributions. The probability $p(x_\nu|c_k)$ of each point x_ν , given that the class is c_k , is calculated as follows:

$$p(x_\nu | c_k) = \frac{M}{n} \frac{1}{N_k h} \sum_{j=1}^{N_k} K\left(\frac{x_\nu - x_j}{h}\right), \quad (8)$$

where N_k is the number of data items belonging to class c_k , h is a smoothing parameter called bandwidth, and $K(\xi)$ is a kernel function to choose among normalized, non-negative and symmetric functions.

3. Probability of each class, in correspondence of grid points, is calculated by using Bayes' theorem:

$$p(c_k | x_\nu) = \frac{p(x_\nu | c_k) P(c_k)}{\sum_{i=1}^C p(x_\nu | c_i) P(c_i)}, \quad (9)$$

where prior probabilities $P(c_k)$ can be calculated by using the training dataset:

$$P(c_k) = \frac{N_k}{N} \quad (10)$$

Hence, non-parametric functions describing posterior probabilities are calculated.

3.2 Transformation into Interpretable Fuzzy Sets

This step represents the main novelty of the approach, linking constrained fuzzy partitions and posterior probabilities. The assumption is made here that any non-parametric function describing posterior probability of a class can be approximated by a linear combination of parametric membership functions of a number T of normal, orthogonal, and convex fuzzy sets, which model terms of a linguistic variable:

$$p(c_k | x) \cong \sum_{t=1}^T \lambda_{t-k} \mu_t(x), \quad (11)$$

In our proposal, membership functions are generated by using sigmoid functions:

$$\begin{cases} \mu_1(x) = 1 - \frac{1}{1 + \exp(a_1x + b_1)} \\ \mu_t(x) = \frac{1}{1 + \exp(a_{t-1}x + b_{t-1})} - \frac{1}{1 + \exp(a_t x + b_t)}, \\ \mu_T(x) = \frac{1}{1 + \exp(a_{T-1}x + b_{T-1})} \end{cases} \tag{12}$$

where $t \in \{2, \dots, T-1\}$. It is simple to verify that these fuzzy sets are orthogonal and convex. They are also approximately normal, if parameters a_t and b_t are properly constrained, in order to obtain, with ε as little as desired:

$$\forall t \in \{1, \dots, T\}, \begin{cases} 0 < \min\{\mu_t(x), x \in U\} < \varepsilon \\ 1 - \varepsilon < \max\{\mu_t(x), x \in U\} < 1 \end{cases} \tag{1'}$$

The problem of approximating $p(c_k|x)$ by using (11) is an optimization problem which can be solved by any known algorithm. The parameters to optimize are a_t , b_t , and $\lambda_{t,k}$, with $t=1, \dots, T$, and $k=1, \dots, C$. Noticing that $\lambda_{t,k}$ must satisfy the condition:

$$\forall t, \sum_{k=1}^C \lambda_{t-k} = 1, \tag{13}$$

then the number of parameters to optimize is $2T+T(C-1)$, which should be sufficiently lower than N . The number T of linguistic terms is chosen by starting from $T = 2$ and then adding one term at a time if the root mean square error of the approximation exceeds a threshold ERR_{\max} . The last added term is removed if ERR_{\max} does not decrease significantly (at least of a certain percentage $ErrDec\%$). No more term is added if $T = T_{\max}$. The choice of $T_{\max} \leq 9$ is recommended for interpretability. Once T has been found, each term is labeled using linguistic terms chosen in the scale $\{extremely\ low, \textit{very low, low, fairly low, medium, fairly high, high, very high, extremely high}\}$.

As a result, for each variable, an interpretable fuzzy partition (12) is obtained, and probability can be calculated as a linear combination of membership grades (11).

3.3 Rule Base Construction and Inference

The proposed approach allows constructing a rule base made of the following set of very simple rules:

$$\forall t \in \{1, \dots, T\}, \forall k \in \{1, \dots, C\}, r_{t-k}(\lambda_{t-k}): \textit{If } x \textit{ is } F_t \textit{ then } y = c_k, \tag{14}$$

each of them associated with the respective weight λ_{t-k} , where each fuzzy set F_t is defined by one of (12).

In order to present the rule base in a more compact form, (14) can be written as:

$$\forall t \in \{1, \dots, T\}, r_t: \text{If } x \text{ is } F_t \text{ then } y = \begin{cases} \lambda_{t-1} & c_1 \\ \dots & \\ \lambda_{t-C} & c_C \end{cases}, \quad (14')$$

where each λ_{t-k} can be seen as the probability of having the class c_k , given that the fuzzy set is F_t .

Bounded sum is used as S-norm (strong disjunction in Łukasiewicz fuzzy logic) for aggregating activations of rules which have the same consequence. As a result, given a data item to be classified, the system calculates an activation of each class equal to the probability of the same class, as can be seen by considering (5), (4) and (11):

$$A_k(x) = \min \left[\sum_{i=1}^R \delta_{i,k} \alpha_i, 1 \right] = \min \left[\sum_{t=1}^T \lambda_{t-k} \mu_t(x), 1 \right] \equiv P(c_k | x). \quad (15)$$

As a result, a transparent and interpretable fuzzy system is constructed which allows calculating class probabilities of incoming data items. Moreover, since the results of this approach differ from Bayesian methods only for making a different approximation of real probabilities, the classification power of the two should be very similar, if measured in terms of CR. On the other hand, in our proposal, a “winner takes all” strategy is not considered, but all classes which have a non-zero grade of activation are taken into account. Therefore, the inference of the proposed approach gives results in terms of a set of classes with respective probabilities, while Bayesian method gives results only in terms of the most probable class. In order to evaluate the improvement connected with this characteristic of the proposed approach, the SCE can be considered, which takes into account that high (low) confidence is desired to be associated to right (wrong) solutions.

4 Application to Multiple Sclerosis Lesions

The proposed method has been applied for the construction of a real DSS with the aim of classifying MSLs, using an experimental training dataset, made of four variables referred to lesions identified in Magnetic Resonance images of brain: WM is the fraction of white matter surrounding a tissue, SF is a shape factor, DF is a distance factor measuring differences of colors, and VN is the volumetric dimension of the lesion expressed in number of voxels. Each pattern is labeled, since the classification of the related lesion as normal brain tissue (NBT) or white matter potential lesion (WMPL) is known. The dataset is made of 939 NBT cases and 1905 WMPLs.

By means of the procedure explained in Section 3.1, with a grid number $n=100$, standard Gaussian kernel functions, a bandwidth $h=M/25$, where M is the measure of

U , the posterior probabilities $p(c_{NBT}|x)$ and $p(c_{WMPL}|x)$ of classes NBT and WMPL were found as non-parametric functions of each input variable x , as shown in Fig. 1. Approximation of these functions was made by using the procedure explained in Section 3.2, with $\epsilon = 0.001$. Approximated functions are also shown in Fig. 1.

At the same time, a fuzzy partition was obtained for each of the input variables, as shown in Fig. 2. A different number of linguistic terms was found for each variable: three terms (*low*, *medium*, and *high*) for WM and SF, only two (*low* and *high*) for DF and VN. These numbers were obtained by setting the approximation thresholds $ERR_{max} = 0.02$, $ErrDec\% = 10\%$, and the maximum number of terms $T_{max} = 9$.

For each variable, a separate fuzzy rule base was extracted as explained in Section 3.3. In Table 1, the weights of the rules of all the rule bases are reported. Each rule base can be used to make inference by using bounded sum S-norm for aggregation, and obtaining as a result good approximations of probabilities of a lesion to belong to classes NBT and WMPL, given a value of one of the input variables. The results show that the best variable to be used is SF, which allows to obtain a CR of 73.6%, while other CRs are: 70.3 for WM, 68.2 for DF, and 67.0 for VN. The best rule base is presented in the compact structure (14'), taking into account the weights of Table 1:

$$\begin{array}{ll}
 r_{low}: & \text{If SF is low then WMPL} \\
 r_{medium}: & \text{If SF is medium then } \left\{ \begin{array}{l} 0.39 \text{ NBT} \\ 0.61 \text{ WMPL} \end{array} \right. \\
 r_{high}: & \text{If SF is high then NBT}
 \end{array}$$

The CR obtained by this rule base is very similar to the CR of the correspondent Bayesian classifier. In particular, both methods show in this case similar sensitivity (~0.97) and specificity (~0.26). On the other hand, a SCE of 0.17 is obtained by using the proposed method, while a worse SCE of 0.36 is obtained by Bayesian classifier.

5 Conclusions

In this paper, a novel hybrid approach based on fuzzy logic and probabilistic view was proposed, in order to combine the good performances of a statistical approach with the high interpretability of a fuzzy system based on the partition of variables into linguistic terms. This was achieved by approximating non-parametric functions describing posterior probabilities with linear combination of parametric membership functions, describing terms of linguistic variables. As a result, the partition of variables and the rule base were constructed, thus obtaining an interpretable fuzzy system which well approximates probabilities of different classes, given a value of an input variable.

The approach is described in order to construct one-dimensional rule bases, while a multidimensional delineation is left for a future extension.

The reliability of the method was shown by applying it to an experimental database, regarding Multiple Sclerosis Lesions classification.

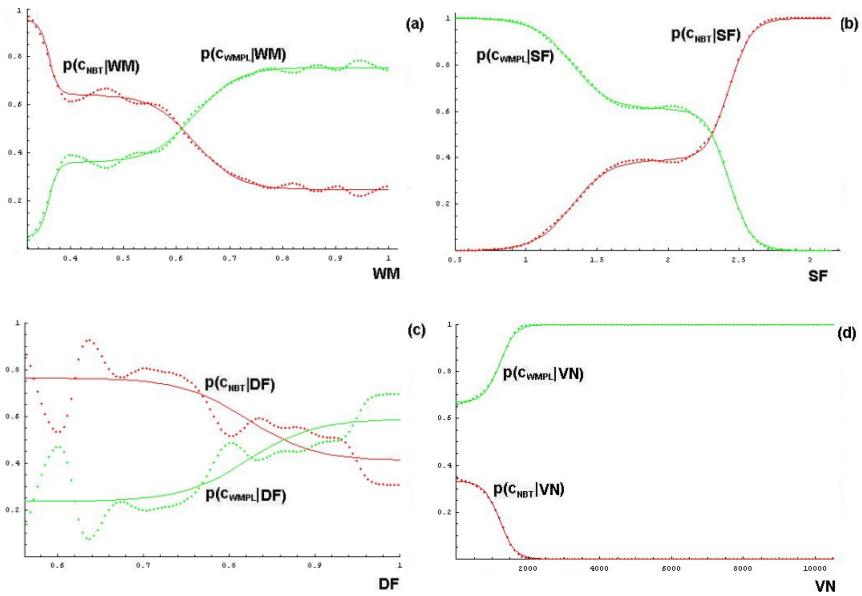


Fig. 1. Probabilities of classes NBT (in red) and WMPL (in green) as functions of x , and their approximations (continuous line), where x is: (a) WM; (b) SF; (c) DF; (d) VN

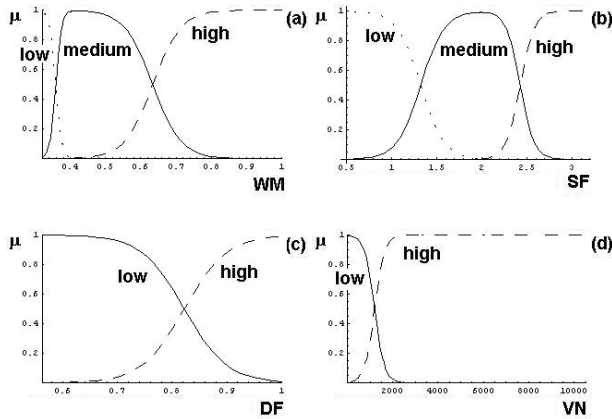


Fig. 2. Membership functions, generated by using sigmoid functions, of fuzzy sets representing partitions of input variable: (a) WM; (b) SF; (c) DF; (d) VN

Table 1. Weights of different rule bases

Variable	$\lambda_{low-NBT}$	$\lambda_{low-WMPL}$	$\lambda_{medium-NBT}$	$\lambda_{medium-WMPL}$	$\lambda_{high-NBT}$	$\lambda_{high-WMPL}$
WM	0.95	0.05	0.64	0.36	0.25	0.75
SF	0	1	0.39	0.61	1	0
DF	0.76	0.24	-	-	0.41	0.59
VN	0.33	0.67	-	-	0	1

The classification power of this approach is very similar to that of Bayesian methods, if measured in terms of classification rate, sensitivity and specificity. However, the proposed approach allows to obtain results in terms of a set of classes with respective probabilities, while classical methods give results only in terms of the most probable class, and this improvement is attested by a lower squared classification error.

References

1. Jacob, S.G., Ramani, R.G.: Mining of classification patterns in clinical data through data mining algorithms. In: Proc. of ICACCI, pp. 997–1003 (2012)
2. Zadeh, L.: Fuzzy sets. *Inform. Control* 8, 338–353 (1965)
3. Palit, A.K., Popovic, D.: *Computational Intelligence in Time Series Forecasting - Theory and Engineering Applications*, pp. 275–303 (2005)
4. Esposito, M., De Falco, I., De Pietro, G.: An evolutionary-fuzzy DSS for assessing health status in multiple sclerosis disease. *Int. J. Med. Inf.* 80, e245–e254 (2011)
5. Guillaume, S.: Designing fuzzy inference systems from data: an interpretability-oriented review. *IEEE Trans. Fuzzy Syst.* 9, 426–443 (2001)
6. Quinlan, J.R.: Induction on decision trees. *Machine Learning* 1, 81–106 (1986)
7. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proc. 5th Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
8. Bezdek, J.C.: *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, Norwell (1981)
9. Yuan, Y., Shaw, M.J.: Induction of fuzzy decision trees. *Fuzzy Sets and Systems* 65, 125–139 (1995)
10. Glorrenec, P.Y.: *Algorithmes d'apprentissage pour systèmes d'inférence floue*. Editions Hermès, Paris (1999)
11. Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. *IEEE Trans. Syst. Man Cybern.* 22, 1414–1427 (1992)
12. Box, G.E.P., Tiao, G.C.: *Bayesian Inference in Statistical Analysis*. Wiley (1973)
13. Akbarzadeh-T., M.-R., Moshtagh-K., M.: A hierarchical fuzzy rule-based approach for aphasia diagnosis. *J. Biomedical Informatics* 40, 465–475 (2007)
14. Schuerz, M., Adlassnig, K.-P., Lagor, C., Schneider, B., Grabner, G.: Definition of fuzzy sets representing medical concepts and acquisition of fuzzy relationships between them by semi-automatic procedures. (Electronic Newsletter) *Fuzzy and Soft Computing Digest* 1(2) (1999)
15. Pota, M., Esposito, M., De Pietro, G.: Transforming probability distributions into membership functions of fuzzy classes: A hypothesis test approach. *Fuzzy Sets and Systems* (2013), <http://dx.doi.org/10.1016/j.fss.2013.03.013>
16. Pota, M., Esposito, M., De Pietro, G.: From likelihood uncertainty to fuzziness: A possibility-based approach for building clinical DSSs. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) *HAIS 2012, Part II. LNCS*, vol. 7209, pp. 369–380. Springer, Heidelberg (2012)
17. Ghazavi, S.N., Liao, T.W.: Medical data mining by fuzzy modeling with selected features. *Artificial Intelligence in Medicine* 43, 195–206 (2008)
18. Guillaume, S., Charnomordic, B.: Learning interpretable fuzzy inference systems with FispPro. *Information Sciences* 181, 4409–4427 (2011)
19. Looftsgaarden, D.O., Quesenberry, C.P.: A non-parametric estimate of a multivariate density function. *The Annales of Mathematical Statistics* 36, 1049–1051 (1965)

An Experimental Case of Study on the Behavior of Multiple Classifier Systems with Class Noise Datasets

José A. Sáez¹, Mikel Galar², Julián Luengo³, and Francisco Herrera¹

¹ Department of Computer Science and Artificial Intelligence, University of Granada, CITIC-UGR, Granada, Spain, 18071

{smja,herrera}@decsai.ugr.es

² Department of Automática y Computación, Universidad Pública de Navarra, Pamplona, Spain, 31006

mikel.galar@unavarra.es

³ Department of Civil Engineering, LSI, University of Burgos, Burgos, Spain, 09006

jluengo@ubu.es

Abstract. Class noise refers to the incorrect labeling of examples in classification datasets and causes the deterioration of the performance of the classifiers. This contribution studies the differences between the behavior of Multiple Classifier Systems and their components when they are trained with data suffering from class noise. The results obtained show that combining classifiers usually help when dealing with classification problems suffering from class noise. In general, the combined models are more accurate than the component classifiers, and the robustness is comparable or slightly better.

Keywords: class noise, multiple classifier systems, classification.

1 Introduction

Data gathered from real-world problems are never perfect and often suffer from noise [18], [16]. In classification problems, two types of noise are distinguished in a dataset: attribute and class noise [19]. Between them, class noise, which is produced when there are incorrectly labeled examples, is usually recognized as the most disruptive for classifier performance [19]. In the literature, two types of class noise are usually distinguished: (i) contradictory examples [5] – duplicate examples having different class labels –, and (ii) misclassifications [19] – examples that are labeled with a class different from the real one. Since misclassifications are the most common in real-world data and they cannot be easily detected as can be done with contradictory examples [19], most of the literature, and also this contribution, is focused on misclassifications, and the term class noise usually refers to this type of noise [17], [15]. The performance of the classifiers built with mislabeled data will depend on the number of examples with an incorrect class labels, but also on the capability of the methods to tolerate that class noise.

Therefore, class noise is a complex problem [18]. Multiple Classifier Systems (MCSs) are usually presented as an alternative to try to improve classification performance in difficult problems [7], [6]. One could ask whether constructing MCSs implies an improvement in classifier performance with respect to individual classifiers dealing with the also difficult problem of training with data suffering from class noise. Combining classifiers enables one to take advantage of the strengths of each method, while avoiding their weaknesses. Thus, if we could use several classifiers which are robust to different types of class noise or mislabeled examples – for example, class noise in the class boundaries or in the core of the class – we could expect that each method could complement each other, avoiding the classification errors in such parts of the classification domain.

This contribution aims to develop an analysis of the behavior of several MCSs with respect to their individual components in the framework of learning with class noise datasets. Forty real-world datasets will be considered and several levels of class noise will be introduced into them following two different schemes: uniform and pairwise [19] - see Section 3 for details. The results taken from these datasets will be analyzed considering two different factors: (i) the performance, and (ii) the robustness. Robustness [8], [15] is the capability of a learning algorithm to build models that are insensitive to data corruptions. Thus, the more robust an algorithm is, the more similar the models built from clean and class noise data. Robustness might be more important than performance on the framework of noisy data, since it allows one to analyze the expected behavior of the algorithm when the level and type of class noise are unknown. This contribution, in contrast to the work of [15] that is focused on ensembles built with the same classification algorithm via a one-versus-one strategy, studies the combination of three heterogeneous classifiers: a decision tree generator (C4.5 [14]), a black-box algorithm (SVM [2]) – which is not considered in [15] – and an instance-based learning method (k -NN [12]).

The rest of this contribution is organized as follows. Section 2 presents MCSs. Next, Section 3 describes the experimental framework. Section 4 includes the experimental results and their analysis. Finally, Section 5 presents some concluding remarks.

2 Multiple Classifier Systems for Classification Tasks

In this section, first MCSs are briefly described in Section 2.1, whereas how the MCSs of this contribution are built is explained in Section 2.2.

2.1 On the Combination of Classifiers

Combining classifiers lets one to avoid the necessity to choose a specific classifier and taking advantage of the strengths of each method. This fact is particularly useful when several classification problems are considered together, since some classifiers may excel in some datasets and perform poorly in others [6], [7].

There are four main strategies to build MCSs: dynamic classifier selection, multi-stage organization, and sequential and parallel approaches [7]. The majority of classifier combination research focuses on the fourth approach, due to its simplicity, and we will also focus on it. In the parallel approach the same input example is submitted to all available classifiers in parallel and the outputs from each classifier are then combined to obtain the final prediction. Many decision combination proposals can be found in the literature, such as the intersection of decision regions [4], voting methods [11], use of the Dempster-Shafer theory [10] or ranking methods [7]. In concrete, we will use the majority vote [11]. This is a simple but powerful approach, where each classifier gives a vote for the predicted class and the most voted one is chosen as the output.

2.2 Multiple Classifier Systems Used in the Experimentation

The choice of the learning algorithms used in this contribution were selected because these methods have a highly differentiated and well known noise-robustness, which is important in order to properly evaluate the performance of MCSs in the presence of class noise data. These methods are:

- C4.5 [14], which is considered a robust learner tolerant to noisy data since it uses pruning strategies to reduce the chances of classifiers being affected by noisy examples from the training data. The default parameters are used for C4.5 in our experiments, considering a post-prune process.
- *Support Vector Machine* (SVM) [2] relies on the support vectors, which are identified from the training instances, to derive the decision model. Thus, the hyperplanes found by SVM can be easily altered including or excluding a single noisy example [13]. SVM is executed with the following parameters: $cost = 100$, $tolerance = 0.001$, $\epsilon = 10^{-12}$, $kernel = Puk$ with $\sigma = 1$, $\omega = 1$.
- *k-Nearest Neighbors* (k -NN) [12]. Three different values of k are considered: 1, 3 and 5. A higher value of k determines a lower sensitivity to noise [9].

Three values of k for k -NN are considered (1, 3 and 5), so we will create three different MCS- k denoted as MCS-1, MCS-3 and MCS-5. Thus, each MCS- k will contain C4.5, SVM and k -NN with the selected k as base classifiers with a different noise-robustness.

3 Experimental Framework

First, this section describes the processes to induce class noise into original base datasets (Section 3.1). Then, the methodology for the analysis of the results is explained in Section 3.2.

3.1 Introducing Class Noise into Datasets

The experimentation is based on forty real-world classification problems from the KEEL-dataset repository¹ [1]. Table 1 shows the datasets with the number

¹ <http://www.keel.es/datasets.php>

Table 1. Base datasets

Dataset	#EX	#AT	#CL	Dataset	#EX	#AT	#CL	Dataset	#EX	#AT	#CL	Dataset	#EX	#AT	#CL
banana	5300	2	2	spambase	4597	57	2	hayes-roth	160	4	3	glass	214	9	7
german	1000	20	2	twonorm	7400	20	2	car	1728	6	4	shuttle	2175	9	7
heart	270	13	2	wdbc	569	30	2	lymph.	148	18	4	zoo	101	16	7
ionosphere	351	33	2	balance	625	4	3	vehicle	846	18	4	satimage	643	36	7
magic	19020	10	2	splice	319	60	3	nursery	1296	8	5	segment	2310	19	7
monk	432	6	2	contracep.	1473	9	3	page-blocks	548	10	5	ecoli	336	7	8
phoneme	5404	5	2	iris	150	4	3	cleveland	297	13	5	led7digit	500	7	10
pima	768	8	2	new-thyroid	215	5	3	automobile	159	25	6	penbased	1099	16	10
ring	7400	20	2	thyroid	720	21	3	dermatology	358	33	6	yeast	1484	8	10
sonar	208	60	2	wine	178	13	3	flare	1066	11	6	vowel	990	13	11

of classes (#CL), the number of examples (#EX) and the number of attributes (#AT). In these datasets the initial amount of class noise present is unknown. Therefore, no assumptions about the noise level can be made. In order to control the amount of noise in each dataset, different class noise levels $x\%$ are introduced in a supervised manner with the following two schemes:

1. **Uniform class noise** [17]. $x\%$ of the examples are corrupted. The class labels of these examples are randomly replaced by another one.
2. **Pairwise class noise** [19]. Let X be the majority class and Y the second majority class, an example with the label X has a probability of $x/100$ of being incorrectly labeled as Y .

The accuracy estimation of the classifiers in a dataset is obtained by means of 5 runs of a stratified 5-fold cross-validation. 5 partitions are used because, if each partition has a large number of examples, the noise effects will be more notable, facilitating their analysis. New class noise datasets will be created from the aforementioned forty base datasets, considering the noise levels ranging from $x = 0\%$ (base datasets) to $x = 40\%$, by increments of 10%. Only the training partitions are induced with noise whereas the test partitions remain unaltered.

3.2 Methodology of Analysis

In order to check the behavior of the different methods when dealing with class noise, the results of each MCS are compared with those of their individual components using two distinct properties:

1. The performance of the classification algorithms on the clean test partitions for each level of induced class noise in the training partition, defined as its accuracy rate. For the sake of brevity, only averaged results are shown, even though our conclusions are based on the proper statistical analysis, which considers all the results.

2. The robustness of each method is estimated with the *relative loss of accuracy* (RLA) [15] (Equation 1), which is used to measure the percentage of variation of the accuracy of the classifiers at a concrete class noise level with respect to the original case with no additional noise:

$$RLA_{x\%} = \frac{Acc_{0\%} - Acc_{x\%}}{Acc_{0\%}}, \quad (1)$$

where $RLA_{x\%}$ is the relative loss of accuracy at a noise level $x\%$, $Acc_{0\%}$ is the test accuracy in the original case, that is, with 0% of induced noise, and $Acc_{x\%}$ is the test accuracy with a noise level $x\%$. Therefore, the performance of the classifiers learned with the base dataset will be compared with the performance of the classifiers learned using the class noise dataset. Both performance and robustness are studied because the conclusions reached with one of these metrics need not imply the same conclusions with the other.

In order to properly analyze the performance and RLA results, Wilcoxon's statistical test is used [3]. This is a non-parametric pairwise test that aims to detect significant differences between two sample medians. For each class noise type and level, the MCS and each single classifier will be compared using Wilcoxon's test and the p-values associated with these comparisons will be obtained. The p-value represents the lowest level of significance of a hypothesis that results in a rejection and it allows one to know whether two algorithms are significantly different and the degree of this difference.

4 Class Noise Data in Multiple Classifier Systems

This section presents the results of performance and robustness of the MCSs trained with class noise data with respect to their individual classifier components. Table 2 shows the performance (left hand) and robustness (right hand) results of each classification algorithm at each noise level on datasets with uniform class noise, whereas Table 3 shows the same information with pairwise class noise. The robustness can only be measured if the noise level is higher than 0%, so the robustness results are presented from a noise level of 10% onwards.

The results in Table 2 comparing each MCS- k against its individual components dealing with uniform class noise are summarized below:

– Performance results with uniform class noise.

- *Comparison with SVM.* MCS- k outperforms SVM. The highest values of k produce lower p-values, being the null hypothesis rejected in all cases.
- *Comparison with C4.5.* In MCS-1 evidences to reject the null hypothesis are only found without noise. In MCS-3 we can only state the same up to 30% noise level. For the rest of the noise levels, for both MCS-1 and MCS-3 show no evidence to reject the null hypothesis compared to C4.5, even though MCS-3 obtains more ranks than C4.5. MCS-5 is better than all its individual components at all noise levels as the null hypothesis are rejected in all cases.

Table 2. Results on datasets with uniform class noise. A star ‘*’ indicates that the single algorithm obtains more ranks than the MCS in Wilcoxon’s test.

		Performance					Robustness				
		0%	10%	20%	30%	40%	10%	20%	30%	40%	
Results	Single meth.	SVM	83.25	79.58	76.55	73.82	70.69	4.44	8.16	11.38	15.08
		C4.5	82.96	82.08	79.97	77.90	74.51	1.10	3.78	6.36	10.54
		1-NN	81.42	76.28	71.22	65.88	61.00	6.16	12.10	18.71	24.15
		3-NN	82.32	80.84	78.18	75.09	70.71	1.85	4.88	8.60	13.59
		5-NN	82.32	81.56	80.09	78.18	75.09	0.95	2.64	4.97	8.31
Results	MCSs	MCS-1	85.42	83.00	80.09	77.10	73.20	2.91	6.40	9.95	14.54
		MCS-3	85.48	84.25	81.98	79.69	76.04	1.47	4.24	6.92	11.20
		MCS-5	85.52	84.57	82.69	80.75	77.56	1.12	3.47	5.78	9.55
p-values	MCS-1	SVM	5.20E-03	1.10E-04	5.80E-05	2.80E-04	7.20E-03	7.20E-03	1.50E-02	8.80E-02	6.60E-01
		C4.5	1.80E-03	3.90E-01	9.5E-01*	3.5E-01*	2.0E-01*	1.5E-06*	3.1E-05*	4.6E-05*	8.2E-05*
		1-NN	7.10E-04	1.30E-07	1.00E-07	6.10E-08	7.00E-08	1.00E-07	1.20E-06	1.80E-07	1.50E-06
p-values	MCS-3	SVM	1.30E-02	1.50E-06	8.10E-07	1.70E-06	1.80E-05	4.60E-05	1.10E-04	1.50E-04	2.80E-03
		C4.5	5.00E-04	3.50E-03	2.00E-02	8.30E-02	1.80E-01	9.3E-02*	2.6E-01*	2.8E-01*	3.0E-01*
		3-NN	3.10E-03	7.80E-04	2.70E-04	1.90E-04	7.80E-05	1.50E-02	4.40E-02	2.80E-02	9.90E-03
p-values	MCS-5	SVM	1.00E-02	2.10E-07	1.70E-07	2.30E-07	2.10E-06	3.10E-06	2.30E-05	8.60E-06	1.70E-04
		C4.5	2.80E-04	4.50E-04	8.60E-04	2.00E-03	5.60E-03	7.10E-01	3.80E-01	3.80E-01	2.50E-01
		5-NN	5.00E-03	1.10E-02	6.20E-02	1.00E-01	9.00E-02	2.7E-01*	2.8E-01*	5.9E-01*	9.7E-01*

Table 3. Results on datasets with pairwise class noise. A star ‘*’ indicates that the single algorithm obtains more ranks than the MCS in Wilcoxon’s test.

		Performance					Robustness				
		0%	10%	20%	30%	40%	10%	20%	30%	40%	
Results	Single meth.	SVM	83.25	80.74	79.11	76.64	73.13	2.97	4.86	7.81	12.01
		C4.5	82.96	82.17	80.87	78.81	74.83	1.00	2.66	5.33	10.19
		1-NN	81.42	77.73	74.25	70.46	66.58	4.20	8.21	12.75	17.20
		3-NN	82.32	80.87	77.81	73.45	68.17	1.59	5.10	9.96	15.93
		5-NN	82.32	81.61	79.33	75.19	69.25	0.78	3.33	7.80	14.53
Results	MCSs	MCS-1	85.42	83.95	82.21	79.52	75.25	1.73	3.86	7.11	12.08
		MCS-3	85.48	84.61	83.06	80.24	75.61	1.05	2.97	6.35	11.83
		MCS-5	85.52	84.86	83.50	80.76	75.93	0.81	2.51	5.81	11.52
p-values	MCS-1	SVM	5.20E-03	1.20E-04	2.00E-04	2.10E-03	3.80E-02	6.60E-03	7.20E-02	3.80E-01	5.50E-01
		C4.5	1.80E-03	5.80E-02	3.50E-01	8.20E-01	8.0E-01*	2.0E-04*	1.7E-03*	2.7E-03*	7.8E-03*
		1-NN	7.10E-04	1.20E-07	8.20E-08	5.60E-08	4.50E-08	6.70E-06	1.00E-05	6.30E-06	4.40E-05
p-values	MCS-3	SVM	1.30E-02	6.20E-05	3.30E-05	2.70E-04	1.90E-02	3.90E-04	3.70E-03	6.40E-02	4.40E-01
		C4.5	5.00E-04	2.30E-03	1.90E-02	4.00E-01	9.7E-01*	2.4E-01*	9.0E-02*	7.4E-02*	5.6E-03*
		3-NN	3.10E-03	4.30E-05	3.80E-07	2.00E-07	1.30E-07	4.20E-03	4.30E-04	7.30E-05	5.50E-05
p-values	MCS-5	SVM	1.00E-02	2.00E-05	1.20E-05	6.90E-05	1.40E-02	6.60E-05	4.10E-04	1.20E-02	3.30E-01
		C4.5	2.80E-04	3.90E-04	1.80E-03	8.80E-02	5.80E-01	5.60E-01	7.4E-01*	3.6E-01*	1.2E-02*
		5-NN	5.00E-03	2.80E-03	6.70E-06	7.10E-07	3.00E-07	4.60E-01	3.00E-03	8.60E-04	6.90E-05

- *Comparison with k-NN.* The null hypothesis can be rejected when comparing sMCS- k and k -NN regardless of the value of k . Furthermore, as the value of k increases, so does the corresponding p-value at the different noise levels (note that very low p-values are obtained in the case of $k = 1$).

– **Robustness results with uniform class noise.**

- *Comparison with SVM.* MCS- k is significantly more robust than SVM, even though MCS-1 and SVM are equivalent from a noise level 30% onwards.
- *Comparison with C4.5.* C4.5 is more robust than MCS-1. MCS-3 and MCS-5 are equivalent to C4.5.
- *Comparison with k -NN.* MCS- k can be considered better than k -NN as the null hypothesis can be rejected, whereas the hypothesis of equivalence between MCS-5 and 5-NN cannot be rejected.

Hereafter, the pairwise class noise results in Table 3 comparing each MCS- k against its individual components are summarized:

– **Performance results with pairwise class noise.**

- *Comparison with SVM.* MCS- k is better than its individual components independently of the value of k . The p-values at the different noise levels decrease as k increases. In all cases Wilcoxon's test is able to reject the null hypothesis.
- *Comparison with C4.5.* Generally, MCS- k obtains more ranks than C4.5 and is better when the noise level is below 10% ($k = 1$), 20% ($k = 3$) and 30% ($k = 5$) since the statistical test find differences between the corresponding accuracy samples.
- *Comparison with k -NN.* MCS- k outperforms k -NN regardless of the value of k , also obtaining p-values lower than 0.5.

– **Robustness results with pairwise class noise.**

- *Comparison with SVM.* MCS-1 overcomes SVM up to a 20% noise level. In the case of MCS-3 and MCS-5, the results provide evidences to reject the null hypothesis in favor of the MCS- k remain up to 30%.
- *Comparison with C4.5.* C4.5 is more robust than MCS-1. MCS-3 and C4.5 are equally robust in most of the cases, even though C4.5 excels at some noise levels. No differences are found between MCS-5 and C4.5.
- *Comparison with k -NN.* MCS- k is more robust than k -NN at all the noise levels and the p-values increase with the value of k indicating that the samples support the null hypothesis rejection at all noise levels.

From these results, the following points can be concluded:

1. The performance results show that each MCS- k generally outperforms its single classifier components dealing with *uniform class noise*. MCS- k is better than SVM and k -NN regardless of the value of k , whereas it only performs better than C4.5 at the lowest noise levels – even though the noise level where MCS- k is better increases together with the value of k , obtaining significant differences at all the noise levels with $k = 5$. Regarding the results with *pairwise class noise*, MCS- k improves on SVM, it is better than C4.5 at the lowest noise levels – these noise levels are lower than those of the uniform class noise – and also outperforms k -NN.

2. The robustness results show that the higher the value of k , the greater the robustness of the MCS- k is. Moreover, MCS- k are generally more robust with *uniform class noise*, even though they are never more robust than all their components. MCS- k are more robust against *uniform class noise* than against *pairwise class noise*. With the *pairwise class noise*, MCS- k are more robust than their single classifiers at lower noise levels than with *uniform noise* (in the case of SVM) or are indeed less robust or equivalent in all the noise levels (in the case of C4.5) – even though MCS- k is generally more robust than k -NN.
3. The *uniform class noise* is the most disruptive class noise. However, from a noise level 10-20% onwards, the *pairwise class noise* becomes more disruptive for 3-NN and 5-NN (and consequently for MCS-3 and MCS-5). This fact clearly indicates that the behavior of each single classification algorithm trained with noisy data influences that of the corresponding MCS into which it is incorporated. The higher disruptiveness of the *uniform class noise* in MCSs can be attributed to the fact that this type of noise affects all the output domain, that is, all the classes, to the same extent, whereas the *pairwise class noise* only affects the two majority classes.

5 Concluding Remarks

This contribution analyzes to what extent three different MCSs composed of C4.5, SVM and k -NN, which are usually presented as a possible solution to improve classification performance in difficult problems, are also suitable to deal with datasets suffering from class noise. The classifier performance and robustness results of the three MCSs considered with respect to their single components over datasets with different types and levels of class noise have been studied.

The uniform class noise scheme seems to be more detrimental for the performance of the classifiers considered in this contribution than the pairwise class noise scheme because mislabeled examples affect all the classes to the same extent with the former, whereas only the two majority classes affected in the latter. Moreover, it has been shown that the behavior of each single classification algorithm (C4.5, SVM and k -NN) trained with class noise data directly influences that of each of the three corresponding MCS into which it is incorporated.

The performance results obtained show that the three MCS studied (MCS-1, MCS-3 and MCS-5) usually produce more accurate solutions than their single components in classification problems with class noise. Generally, the MCSs studied outperform k -NN and SVM, whereas they are only better than C4.5 at the lowest class noise levels – this fact was somehow expected since C4.5 is more tolerant to noise than SVM and k -NN. Moreover, the behavior of each of the three MCS studied is usually better, for the same class noise level, with the uniform class noise scheme.

The robustness results show that the MCSs studied are generally more robust than their single components with the most disruptive class noise scheme, that is, the uniform one. Improvements in the robustness are more difficult to achieve

with the pairwise class noise. Moreover, in the case of the MCSs studied, the higher the value of k in the MCS- k , the greater the robustness of the MCS- k is. From this work future efforts will be focused on analyzing more noise types and the analysis of different MCS models aiming to provide an explanation of the better performance obtained by the MCSs.

Acknowledgment. Supported by the Spanish Ministry of Science and Technology under Projects TIN2011-28488 and TIN2010-15055, and also by the Regional Project P10-TIC-6858. José A. Sáez holds an FPU scholarship from the Spanish Ministry of Education and Science.

References

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Journal of Multiple-Valued Logic and Soft Computing* 17(2-3), 255–287 (2011)
2. Cortes, C., Vapnik, V.: Support vector networks. *Machine Learning* 20, 273–297 (1995)
3. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
4. Haralick, R.M.: The table look-up rule. *Communications in Statistics - Theory and Methods* A5(12), 1163–1191 (1976)
5. Hernández, M.A., Stolfo, S.J.: Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem. *Data Mining and Knowledge Discovery* 2, 9–37 (1998)
6. Ho, T.K.: Multiple Classifier Combination: Lessons and Next Steps. In: Kandel, Bunke, E. (eds.) *Hybrid Methods in Pattern Recognition*, pp. 171–198. World Scientific (2002)
7. Ho, T.K., Hull, J.J., Srihari, S.N.: Decision Combination in Multiple Classifier Systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(1), 66–75 (1994)
8. Huber, P.J.: *Robust Statistics*. John Wiley and Sons, New York (1981)
9. Kononenko, I., Kukar, M.: *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Horwood Publishing Limited (2007)
10. Mandler, E., Schuermann, J.: Combining the classification results of independent classifiers based on the Dempster/Shafer theory of evidence. In: Gelsema, E.S., Kanal, L.N. (eds.) *Pattern Recognition and Artificial Intelligence*, pp. 381–393. North-Holland, Amsterdam (1988)
11. Mazurov, V.D., Krivonogov, A.I., Kazantsev, V.S.: Solving of optimization and identification problems by the committee methods. *Pattern Recognition* 20, 371–378 (1987)
12. McLachlan, G.J.: *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley and Sons (2004)
13. Nettleton, D., Orriols-Puig, A., Fornells, A.: A Study of the Effect of Different Types of Noise on the Precision of Supervised Learning Techniques. *Artificial Intelligence Review* 33, 275–306 (2010)
14. Quinlan, J.R.: *C4.5: programs for machine learning*. Morgan Kaufmann Publishers, San Francisco (1993)

15. Sáez, J.A., Galar, M., Luengo, J., Herrera, F.: Analyzing the presence of noise in multi-class problems: alleviating its influence with the One-vs-One decomposition. *Knowledge and Information Systems* (in press, 2013), doi:10.1007/s10115-012-0570-1
16. Sáez, J.A., Luengo, J., Herrera, F.: Predicting Noise Filtering Efficacy with Data Complexity Measures for Nearest Neighbor Classification. *Pattern Recognition* 46(1), 355–364 (2013)
17. Teng, C.M.: Correcting Noisy Data. In: *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 239–248. Morgan Kaufmann Publishers, San Francisco (1999)
18. Zhong, S., Khoshgoftaar, T.M., Seliya, N.: Analyzing Software Measurement Data with Clustering Techniques. *IEEE Intelligent Systems* 19(2), 20–27 (2004)
19. Zhu, X., Wu, X.: Class Noise vs. Attribute Noise: A Quantitative Study. *Artificial Intelligence Review* 22, 177–210 (2004)

A Sensitivity Analysis for Quality Measures of Quantitative Association Rules

María Martínez-Ballesteros¹, Francisco Martínez-Álvarez²,
Alicia Troncoso², and José C. Riquelme¹

¹ Department of Computer Science, University of Seville, Spain
{[mariamartinez](mailto:mariamartinez@us.es), [riquelme](mailto:riquelme@us.es)}@us.es

² Department of Computer Science, Pablo de Olavide University of Seville, Spain
{[fmaralv](mailto:fmaralv@upo.es), [ali](mailto:ali@upo.es)}@upo.es

Abstract. There exist several fitness function proposals based on a combination of weighted objectives to optimize the discovery of association rules. Nevertheless, some differences in the measures used to assess the quality of association rules could be obtained according to the values of such weights. Therefore, in such proposals it is very important the user's decision in order to specify the weights or coefficients of the optimized objectives. Thus, this work presents an analysis on the sensitivity of several quality measures when the weights included in the fitness function of the existing QARGA algorithm are modified. Finally, a comparative analysis of the results obtained according to the weights setup is provided.

Keywords: Data mining, sensitivity analysis, quantitative association rules, quality measures.

1 Introduction

The use of efficient computational techniques has become a task of the utmost importance due to the high volume of data that can be stored nowadays. In this context, the discovery of association rules (AR) –and particularly of quantitative association rules (QAR) in this work– is a popular and well-known methodology to discover significant and apparently hidden relations among variables that form databases [2]. This discovery of knowledge is based on statistical techniques such as correlation analysis and variance. One of the most used and cited algorithms is Apriori [1].

When the domain is continuous, the AR are known as QAR. In this context, let $F = \{F_1, \dots, F_n\}$ be a set of features, with values in \mathbb{R} . Let A and C be two disjoint subsets of F , that is, $A \subset F$, $C \subset F$, and $A \cap C = \emptyset$. A QAR is a rule $X \Rightarrow Y$, in which features in A belong to the antecedent X , and features in C belong to the consequent Y , such that X and Y are formed by a conjunction of multiple boolean expressions of the form $F_i \in [v_1, v_2]$. The consequent Y is usually a single expression.

The AR extraction process is a non-supervised learning technique to explore data properties. The main goal pursuit is, then, to find groups of attributes

appearing frequently together in a dataset, so to provide comprehensive rules able to explain the existing relations among them. The mining process of AR is usually modeled as a multi-objective problem in which several quality measures of AR are the objectives to be optimized since there not exist an unique measure to determine the AR quality. There are several approaches to solve multi-objective problems. The most common approaches are focused in Pareto-based multi-objective algorithms which try to find the best trade-off between two or more conflicting objectives. However, many others are based in weighted sum fitness functions which formulate the problem as a single-objective optimization problem using parameters of scalarization. Such weighted sum fitness functions allow to find solutions according to the user preferences and emphasize some objectives over others. Most of the existing techniques to discover AR are typically focused in using the support and confidence measures as objectives to be optimized by a weighted sum fitness function. Therefore, the main goal of this work is to conduct an analysis on the sensitivity of such quality measures when the weights in the fitness function vary. Nonetheless, there also exist other measures widely used for both evaluation and optimization of AR [9]. Some of such quality measures are described in Table 1. Note that $n(X)$ is the number of occurrences of the itemset X in the dataset and N is the total number of instances in the dataset. ND stands for negatively dependent, PD for positively dependent and I for independent.

Table 1. Quality measures for quantitative association rules

Measures	Equation	Description	Range
$Sup(X)$	$n(X)/N$	Coverage of X	[0, 1]
$Sup(X \Rightarrow Y)$	$n(X \cap Y)/N$	Generality of the rule	[0, 1]
$Conf(X \Rightarrow Y)$	$sup(X \Rightarrow Y)/sup(X)$	Reliability of the rule	[0, 1]
$Lift(X \Rightarrow Y)$	$sup(X \Rightarrow Y)/(sup(X) \cdot sup(Y))$	Interest of the rule <ul style="list-style-type: none"> • Value < 1: X and Y (ND) • Value = 1: X and Y (I) • Value > 1: X and Y (PD) 	[0, +∞)
$Gain(X \Rightarrow Y)$	$conf(X \Rightarrow Y) - sup(Y)$	Implication of the rule	[-0.5, 1]
$Certainty\ Factor(X \Rightarrow Y)$	<ul style="list-style-type: none"> • If $conf(X \Rightarrow Y) > sup(Y)$: $(conf(X \Rightarrow Y) - sup(Y))/(1 - sup(Y))$ • If $conf(X \Rightarrow Y) \leq sup(Y)$: $(conf(X \Rightarrow Y) - sup(Y))/sup(Y)$ 	Gain normalized <ul style="list-style-type: none"> • Value < 0: X and Y (ND) • Value = 0: X and Y (I) • Value > 0: X and Y (PD) 	[-1, 1]
$Leverage(X \Rightarrow Y)$	$sup(X \Rightarrow Y) - sup(X)sup(Y)$	Strength of the rule <ul style="list-style-type: none"> • Value < 0: X and Y (ND) • Value = 0: X and Y (I) • Value > 0: X and Y (PD) 	[-0.25, 0.25]
$Accuracy(X \Rightarrow Y)$	$sup(X \Rightarrow Y) + sup(\neg X \Rightarrow \neg Y)$	Veracity of the rule	[0, 1]

Thus, we aim to provide guidelines to set the weights of the fitness function according to the objectives to be satisfied by the rules. On the other hand, we intend to establish multiple relationships between the quality measures and variations in the weights of the fitness function by means of the results obtained by the QARGA algorithm [8].

The remainder of the paper is as follows. Section 2 describes some techniques which included a weighted sum fitness function. The QARGA algorithm used in

the study performed and the experimental setup is detailed in Section 3. Section 4 presents and discusses the results obtained by QARGA using different weights in the fitness function. Finally, Section 5 summarizes the conclusions drawn from the analysis conducted.

2 Related Work

There exist several fitness functions proposals based on a combination of weighted objectives in a single equation. Hence, their performance is very sensitive to the choice of the weights of the measures included within the fitness function. Actually, many algorithms to discover AR can be found in the literature. Most of them are based on the methods proposed by Agrawal et al. [1] but such methods require high computational cost and memory. Genetic algorithms, colony algorithms, evolutionary algorithms (EA) and particle swarm algorithms are usually used to overcome such drawbacks. Techniques based on EA have been extensively used for the optimization and adjustment of models in data mining. Evolutionary computation is usually used to discover AR in both EA and genetic programming since they offer a set of advantages for knowledge extraction and specifically for rule induction processes.

A wide range of methods have been proposed to address the discovery and optimization of AR by a weighted sum fitness function. This kind of fitness function has been applied into several optimization problems. For instance, the authors in [7] examined the effect of using weighted sum fitness functions for parent selection and generation update. Such an effect was tested on the performance of NSGA-II for a high-dimensional space of a multi-objective problem.

The authors in [11] proposed an EA-based approach capable of obtaining an undetermined number of quantitative attributes in the antecedent of the rule. Their approach, called GENAR, optimized a weighted fitness function based on the support and confidence measures and the number of recovered instances. The same quality measures plus the comprehensibility and the amplitude of the intervals forming the rule were included in the weighted sum fitness function of the GAR-plus algorithm [12].

In [2] a GA is proposed as a search strategy for both positive and negative QAR mining within databases. The discovery of QAR was optimized by a weighted sum fitness function composed of support, confidence, number of attributes and amplitude. Later, the same authors proposed a multi-objective Pareto-based EA called MODENAR in [3]. Those measures and the recovered records were included within the fitness function to be optimized in such works.

A genetic algorithm was proposed in [14] which optimized support, confidence, comprehensibility and interest of AR included in a weighted fitness function. A weighted support based on the individual weight of the items according to their importance in the dataset was calculated in [13].

3 Methodology

3.1 Description of QARGA

This section describes the main features of the QARGA algorithm, which is used to perform the fitness function sensitivity study according to the weighting of the measures. QARGA is a real-coded genetic algorithm designed to discover existing relationships, specifically QAR, among several variables. A detailed description of the algorithm can be found in [10].

The fitness of each individual in the evolutionary process allows determining which are the best candidates to remain in subsequent generations. In order to make this decision, its calculation involves several measures that provide information about the rules. The fitness function has been designed to maximize a combination of different measures of AR.

The fitness function proposed in [8] to be maximized by QARGA was:

$$f(rule) = w_s \cdot sup + w_c \cdot conf + w_n \cdot nAttrib - w_a \cdot ampl - w_r \cdot recov \quad (1)$$

where *sup* is the support of the rule, *conf* is the confidence of the rule, *recov* is the ratio of instances which had already been covered, *nAttrib* is the number of attributes appearing in the rule and *ampl* is the average size of intervals of the attributes belonging to the rule. Moreover, the fitness function was provided with a set of weights (w_s, w_c, w_n, w_a and w_r) to drive the process of search of rules and will vary depending on the required rules.

However, the user should be aware of the importance of each measure, in order to specify the weights or coefficient because significant differences in the AR quality measures could be obtained. Next section describes the experimentation framework carried out to assess how the weights included in QARGA's fitness function may influence, in order to provide a guide for the user's decision.

3.2 Experimental Design

It is well known that one of the shortcomings of the weighted sum fitness function is the parametrization of such weights. A fitness function-based sensitivity analysis and a detailed study of some weights are discussed in Section 4 to ascertain the relative influence of each weight on the final results obtained by QARGA. The aim of this study is to analyze the behavior of QARGA to achieve optimal solutions.

Therefore, three sets of experiments have been performed by varying the weights for support and confidence measures included in QARGA's fitness function. The first set of experiments has used a minimum support threshold equal to 0 to obtain all the QAR found by QARGA. The second and third set of experiments have used a minimum support threshold equal to 0.05 and 0.1 respectively to penalize the fitness function of those individuals of the population which do not satisfy the minimum support thresholds, respectively. We aim to force QARGA to learn QAR with the established minimum support.

Different configurations of QARGA have been executed by modifying the weights of the support and confidence measures optimized in the fitness function for each set of experiments. Specifically, the weight values for support and confidence measures, henceforth called w_s and w_c respectively, have been varied from 0 to 1 with increments of 0.1 (11 different values for both). Hence, QARGA was run 363 ($3 \times 11 \times 11$) times in total. To be precise, 363 executions have carried out for each dataset, using them as training data. Note that the rest of the weights included in the fitness function of QARGA have been set to 1 in order to avoid the influence of such weight in the results and to ensure that the remaining measures are present in the fitness function.

These set of experiments were designed to highlight the main differences in measures performance when the weights of the fitness function are modified according to several minimum support thresholds.

4 Experimental Study

4.1 Datasets Description and Parameters Setup

This section presents the main features of the datasets used in the sensitivity study carried out. Several datasets have been tested from the public BUFA repository [6]. In particular, the thirty-five public datasets from BUFA repository used in [9]. Note that Buying, Country, College, Education, Read and Usnews Colleged have been preprocessed using K-means Imputation method proposed in [5] (available in the KEEL tool [4]) in order to deal with missing values.

As for the values for the main parameters of QARGA, it is noteworthy that these values have been used for each execution to assess the performance of QARGA according to the different values of the weights included in the fitness function.

The main parameters of QARGA are: 100 for the size of the population, 100 for the number of generations, 0.1 for the mutation probability p_{mut} of the individuals and 0.2 for the mutation probability p_{mutgen} of each gene in the individual. The maximum number of attributes which could include both the antecedent and consequent are 10 and 5, respectively. Note that both the antecedent and consequent must contain one attribute at least. QARGA has obtained 100 QAR for each dataset and each setting of the fitness function weights.

4.2 Sensitivity Analysis of the Quality Measures

In this section, the results obtained by QARGA when optimizing the fitness function with variations in the weights of the measures are discussed. Specifically, the results obtained by QARGA, first, when a minimum support threshold is applied and second, when w_s and w_c are modified in the fitness function are compared.

As described in Section 3.2, QARGA has been executed 363 times for each dataset, that is, a total of 12705 executions. In order to perform the parametric

sensitivity study, the average results for the 35 datasets using the same configuration has been calculated. Several interestingness measures have been calculated to assess the quality of the AR obtained by QARGA for each run. In particular, support, confidence, lift, gain, leverage, accuracy, number of attributes, amplitude of the attributes, number of the rules obtained and percentage of covered records have been computed. A detailed explanation of these measures can be found in [8].

Tables 2 and 3 summarize the behavior of the quality measures depending on the minimum support threshold used by QARGA. Note that similar results have been obtained when the minimum support threshold is 0.05 and 0.1, hence, only the results obtained by QARGA with a minimum support threshold equal to 0.05 are shown in Table 3.

Each table presents the studied quality measures grouped by their performance when the weights associated with the support and confidence measures in the fitness function are increased or decreased.

Table 2. Performance of quality measures according to the support and confidence weights with minimum support equal to 0

Weight	Quality measures grouped by similar behavior					
	Support Leverage Covered instances	Confidence Gain #Rules	Accuracy	Lift	#Attributes	Amplitude
Support ↑	↑	=	=	=	↓	↑
Confidence ↑	=	>0.1 = <0.1 ↑	>0.1 ↑ <0.1 ↓	↑	↑	↓

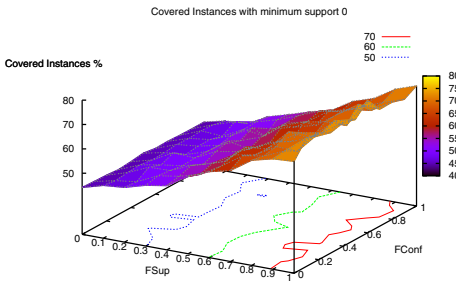


Fig. 1. Covered instances with minimum support 0

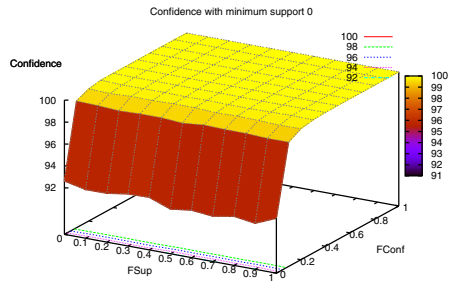


Fig. 2. Confidence with min. sup. 0

Table 2 shows the ten studied quality measures arranged into six groups. It can be noted that w_s is positively correlated with support, leverage, covered instances and amplitude whereas is negatively correlated with the number of attributes if no minimum support threshold is applied. The performance of the other measures under study is not affected by the variations of w_s . With respect to w_c , some differences can be observed. For instance, although support, leverage and covered instances are dependent of w_s , such measures are not influenced by

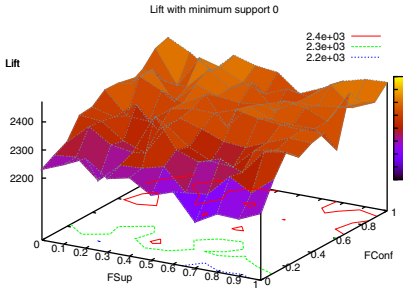


Fig. 3. Lift with minimum support 0

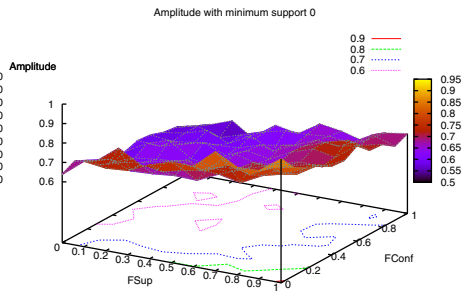


Fig. 4. Amplitude with minimum support 0

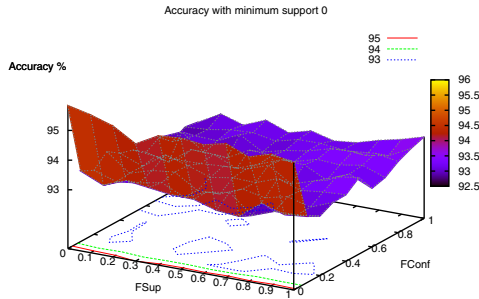


Fig. 5. Accuracy with minimum support 0

w_c . Lift measure and number of attributes are positively correlated with w_c and amplitude is negatively correlated. However, confidence, gain and the number of rules are only increased when w_c achieves values of 0 and 0.1. A w_c greater than 0.1 does not cause alterations in the performance of such measures. It can be observed an opposite behavior in the accuracy since it is negatively correlated with the confidence if such weight is 0 or 0.1 and positively correlated if w_c is greater than 0.1.

Figures 1, 2, 3, 4 and 5 summarize the values obtained for each group of measures when the minimum support threshold is 0. Note that only one measure of each group is displayed due to the similar performance among the measures of each group and space limitations. Figure 1 represents the support, confidence and covered instances measures. It can be observed that their values form an increasing inclined plane relative to w_s . Figure 2 visualizes the values obtained for the confidence measure and its behavior can be extended to the gain measure and number of rules. These measures present an awning model reaching their highest values when w_c is greater than 0.1. Figures 3 and 4 show the lift and amplitude values respectively when the minimum support threshold is 0. These measures do not follow any specific behavior pattern and can be considered as rough models. Finally, the accuracy measure is displayed in Figure 5. It achieves the highest value when w_c is 0. The performance of this measure can be considered as a valley model.

Table 3. Performance of quality measures according to the support and confidence weights with minimum support equal to 0.05

Weight	Quality measures grouped by similar behavior		
	Support Confidence Leverage Amplitude	Lift Accuracy Gain #Attributes	Covered instances #Rules
$minsup = 0.05$			
Support ↑	=	=	<0.8 ↑ ≥0.8 =
Confidence ↑	↑	↓	≤0.3 ↑ >0.3 =

Table 3 displays the ten measures under study grouped into only three groups when a minimum support threshold is applied. It can be appreciated that the performance of these measures are completely different when a minimum support threshold is not applied. For instance, the group composed of support, confidence, leverage and amplitude and the group formed by lift, accuracy, gain and number of attributes are only affected by w_c . These groups are positively and negatively correlated respectively with w_c . Regarding the third group, that is, covered instances and number of rules are only affected when w_s is less than 0.8 and w_c is less or equal to 0.3, both positively correlated. Weights above these values do not cause performance variations on such measures.

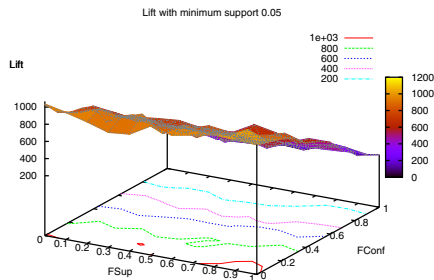
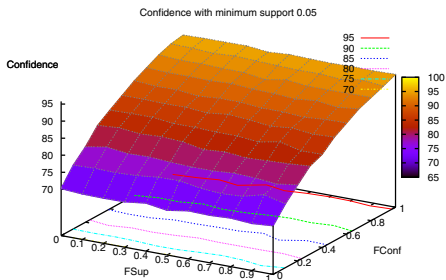


Fig. 6. Confidence with minimum support 0.05 **Fig. 7.** Lift with minimum support 0.05

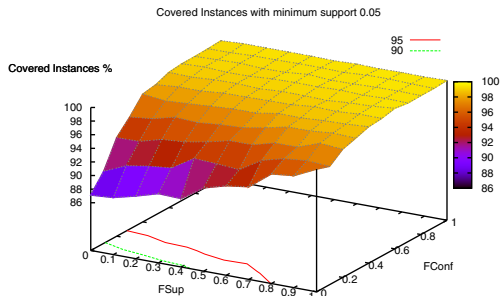


Fig. 8. Covered instances with minimum support 0.05

Figures 6, 7 and 8 illustrate the values obtained for each group of measures when the minimum support threshold is 0.05.

Note that a similar behavior has been obtained when the minimum support threshold is 0.1. Figure 6 represents the performance of confidence, support, leverage and amplitude measures. These measures reach their highest values when w_c is 1. In this case, the confidence behaves as an increasing inclined plane with respect to w_c instead of presenting an awning model as Figure 2. Figure 7 shows the values obtained for the lift measure and summarizes the behavior of accuracy, gain and number of attributes in addition to lift. In this case, these measures get their highest values when w_c is 0 and perform as a decreasing inclined plane relative to w_c . Finally, Figure 8 shows the values obtained for the covered instances. This measure exhibits its maximum value when w_s is 1 and w_c is above 0.3. The covered instances present a behavior similar to an awning model.

As final remarks, we provide the following use recommendations. First, the obtained AR are more specific when the minimum support threshold is 0. Therefore, the support and instances covered values are lesser and the number of attributes and accuracy are greater compared to the values obtained when the minimum support threshold is 0.05. Second, although the confidence, gain, accuracy, and lift are better when the minimum support threshold is 0, it is desirable to apply a minimum support threshold in order to avoid support values below 1%. Taking into account such decision, w_s setting is not important in the final results. And third, it has been observed that w_c is the most influential weight. Thus, values of w_c around 0.5 are desirable because not all measures are increased according to w_c .

Finally, we note that it would be interesting to study the rest of weights included in the fitness function of QARGA to analyze their influence on the AR quality measures.

5 Conclusions

An analysis based on the sensitivity of the quality measures based on the variations of the weights included in the fitness function of the QARGA algorithm has been carried out in this paper. Specifically, QARGA has been applied to several public datasets with the aim of studying how its performance is affected according to the choice of the weights. Significant differences have been observed in the results of ten AR quality measures calculated from the AR obtained by QARGA when w_s and w_c were ranged from 0 to 1. However, w_c has been more influential than w_s over the set of AR quality measures studied. Furthermore, several groups of measures have been identified according to their behavior against the weight variations.

Acknowledgments. The financial support from the Spanish Ministry of Science and Technology, project TIN2011-28956-C02-00, and from the Junta de Andalucía, project P11-TIC-7528, is acknowledged.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the International Conference on Very Large Databases, pp. 478–499 (1994)
2. Alatas, B., Akin, E.: An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft Computing* 10(3), 230–237 (2006)
3. Alatas, B., Akin, E., Karci, A.: MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules. *Applied Soft Computing* 8(1), 646–656 (2008)
4. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J.C., Herrera, F.: Keel: A software tool to assess evolutionary algorithms for data mining problems. *Soft Computing* 13(3), 307–318 (2009)
5. Li, D., Deogun, J., Spaulding, W., Shuart, B.: Towards missing data imputation: A study of fuzzy K-means clustering method. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 573–579. Springer, Heidelberg (2004)
6. Guvenir, H.A., Uysal, I.: Bilkent university function approximation repository (2000), <http://funapp.cs.bilkent.edu.tr>
7. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Empirical analysis of using weighted sum fitness functions in NSGA-II for many-objective 0/1 knapsack problems. In: Proceedings of the International Conference on Computer Modelling and Simulation, pp. 71–76 (2009)
8. Martínez-Ballesteros, M., Martínez-Álvarez, F., Troncoso, A., Riquelme, J.C.: An evolutionary algorithm to discover quantitative association rules in multidimensional time series. *Soft Computing* 15(10), 2065–2084 (2011)
9. Martínez-Ballesteros, M., Martínez-Álvarez, F., Troncoso, A., Riquelme, J.C.: Selecting the best measures to discover quantitative association rules. *Neurocomputing* (in press, 2013), doi: <http://dx.doi.org/10.1016/j.neucom.2013.01.056>
10. Martínez-Ballesteros, M., Riquelme, J.C.: Analysis of measures of quantitative association rules. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part II. LNCS, vol. 6679, pp. 319–326. Springer, Heidelberg (2011)
11. Mata, J., Álvarez, J., Riquelme, J.C.: Mining numeric association rules with genetic algorithms. In: Proceedings of the International Conference on Adaptive and Natural Computing Algorithms, pp. 264–267 (2001)
12. Pachón Álvarez, V., Mata Vázquez, J.: An evolutionary algorithm to discover quantitative association rules from huge databases without the need for an a priori discretization. *Expert Systems with Applications* 39(1), 585–593 (2012)
13. Pears, R., Koh, Y.S., Dobbie, G., Yeap, W.: Weighted association rule mining via a graph based connectivity model. *Information Sciences* 218, 61–84 (2013)
14. Soto, W., Olaya-Benavides, A.: A genetic algorithm for discovery of association rules. In: Proceedings of the International Conference of the Chilean Computer Science Society, pp. 289–293 (2011)

Building a Robust Extreme Learning Machine for Classification in the Presence of Outliers

Ana Luiza B.P. Barros^{1,2} and Guilherme A. Barreto²

¹ Department of Computer Science, State University of Ceará
Campus of Itaperi, Fortaleza, Ceará, Brazil
`analuiza@larces.uece.br`

² Department of Teleinformatics Engineering, Federal University of Ceará
Center of Technology, Campus of Pici, Fortaleza, Ceará, Brazil
`guilherme@deti.ufc.br`

Abstract. The Extreme Learning Machine (ELM), recently proposed by Huang *et al.* [6], is a single-hidden-layered neural network architecture which has been successfully applied to nonlinear regression and classification tasks [5]. A crucial step in the design of the ELM is the computation of the output weight matrix, a step usually performed by means of the ordinary least-squares (OLS) method - a.k.a. Moore-Penrose generalized inverse technique. However, it is well-known that the OLS method produces predictive models which are highly sensitive to outliers in the data. In this paper, we develop an extension of ELM which is robust to outliers caused by labelling errors. To deal with this problem, we suggest the use of M -estimators, a parameter estimation framework widely used in robust regression, to compute the output weight matrix, instead of using the standard OLS solution. The proposed model is robust to label noise not only near the class boundaries, but also far from the class boundaries which can result from mistakes in labelling or gross errors in measuring the input features. We show the usefulness of the proposed classification approach through simulation results using synthetic and real-world data.

Keywords: Extreme Learning Machine, Moore-Penrose Generalized Inverse, Pattern Classification, Outliers, M -Estimation.

1 Introduction

In recent years, there have been an ever increasing interest in a class of supervised one-hidden-layered neural network model, generically called Extreme Learning Machine (ELM), in which the input-to-hidden-layer weights are randomly chosen and hidden-to-output-layer are determined analytically. Mainly due its fast learning speed and ease of implementation [5], several authors have been applying the standard ELM network (and sophisticated variants of it) to a number of complex pattern classification and regression problems [1, 4, 13–18].

It should be mentioned, however, that the aforementioned works have not consistently addressed the important issue of model performance in the presence of outliers, with the work of Horata *et al.* [4] being the only exception.

As a matter of fact, in recent years, it has been observed a growing interest in the development of neural network architectures which are robust to outliers, including proposals for designing RBF networks [10,11], echo-state networks [12] and even ELM networks [4].

It is worth emphasizing that all these previous works (no exception!) approached the issue of robustness to outliers for regression-like problems, such as function approximation and time series prediction. However, in many real-world pattern classification problems, the labels provided for the data samples are noisy. There are typically two kinds of noise in labels. Noise near the class boundaries often occurs because it is hard to consistently label ambiguous data points. Labelling errors far from the class boundaries can occur because of mistakes in labelling or gross errors in measuring the input features. Labelling errors far from the boundary comprises a particular category of *outliers* [9].

Thus, in order to allow ELM-based classifiers to handle labelling errors efficiently, in this paper we propose the use of M -estimators [8], a broad framework widely used for parameter estimation in robust regression problems, to compute the weight matrix operator instead of using the ordinary least squares solution. We show through simulations on synthetic and real-world data that the resulting ELM classifier is very robust to this type of outliers. To the best of our knowledge, this is the performance of the ELM network as a *pattern classifier* is evaluated under the presence of outliers.

The remainder of the paper is organized as follows. In Section 2, we briefly review the fundamentals of ELM in the context of pattern classification. Then, in Section 3 we describe the basic ideas and concepts behind the M -estimation framework and introduce our approach to robust supervised pattern classification using ELM. In Section 4 we present the computer experiments we carried out using synthetic and real-world datasets and also discuss the achieved results. The paper is concluded in Section 5.

2 Fundamentals of the ELM

Let us assume that N data pairs $\{(\mathbf{x}_\mu, \mathbf{d}_\mu)\}_{\mu=1}^N$ are available for building and evaluating the model, where $\mathbf{x}_\mu \in \mathbb{R}^{p+1}$ is the μ -th input pattern¹ and $\mathbf{d}_\mu \in \mathbb{R}^K$ is the corresponding target class label, with K denoting the number of classes. For the labels, we assume an 1-of- K encoding scheme, i.e. for each label vector \mathbf{d}_μ , the component whose index corresponds to the class of pattern \mathbf{x}_μ is set to “+1”, while the other $K - 1$ components are set to “-1”.

Then, let us randomly select N_1 ($N_1 < N$) data pairs from the available data pool and arrange them along the columns of the matrices \mathbf{D} and \mathbf{X} as follows:

$$\mathbf{X} = [\mathbf{x}_1 \mid \mathbf{x}_2 \mid \cdots \mid \mathbf{x}_{N_1}] \quad \text{and} \quad \mathbf{D} = [\mathbf{d}_1 \mid \mathbf{d}_2 \mid \cdots \mid \mathbf{d}_{N_1}]. \tag{1}$$

where $\dim(\mathbf{X}) = (p + 1) \times N_1$ and $\dim(\mathbf{D}) = m \times N_1$.

¹ First component of \mathbf{x}_μ is equal to 1 in order to include the bias.

The ELM is a single-hidden layer feedforward network (SLFN), proposed by [6], for which the weights from the inputs to the hidden neurons are randomly chosen, while only the weights from the hidden neurons to the output are analytically determined. Consequently, ELM offers significant advantages such as fast learning speed, ease of implementation, and less human intervene when compared to more traditional SLFNs, such as the MLP and RBF networks. For a network with p input units, q hidden neurons and C outputs, the i -th output at time step k , is given by

$$o_i(k) = \beta_i^T \mathbf{h}(k), \tag{2}$$

where $\beta_i \in \mathbb{R}^q$, $i = 1, \dots, C$, is the weight vector connecting the hidden neurons to the i -th output neuron, and $\mathbf{h}(k) \in \mathbb{R}^q$ is the vector of hidden neurons' outputs for a given input pattern $\mathbf{x}(k) \in \mathbb{R}^p$. The vector $\mathbf{h}(k)$ itself is defined as

$$\mathbf{h}(k) = [f(\mathbf{w}_1^T \mathbf{x}(k) + b_1), \dots, f(\mathbf{w}_q^T \mathbf{x}(k) + b_q)]^T, \tag{3}$$

where b_l , $l = 1, \dots, q$, is the bias of the l -th hidden neuron, $\mathbf{w}_l \in \mathbb{R}^p$ is the weight vector of the l -th hidden neuron and $f(\cdot)$ is a sigmoidal activation function. Usually, the weight vectors \mathbf{w}_l are randomly sampled from a uniform (or normal) distribution.

Let $\mathbf{H} = [\mathbf{h}(1) \ \mathbf{h}(2) \ \dots \ \mathbf{h}(N)]$ be a $q \times N$ matrix whose N columns are the hidden-layer output vectors $\mathbf{h}(k) \in \mathbb{R}^q$, $k = 1, \dots, N$, where N is the number of available training input patterns. Similarly, let $\mathbf{D} = [\mathbf{d}(1) \ \mathbf{d}(2) \ \dots \ \mathbf{d}(N)]$ be a $C \times N$ matrix whose the k -th column is the target (desired) vector $\mathbf{d}(k) \in \mathbb{R}^C$ associated with the input pattern $\mathbf{x}(k)$, $k = 1, \dots, N$.

Finally, let $\beta = [\beta_1 \ \beta_2 \ \dots \ \beta_C]$ be a $q \times C$ matrix, whose i -th column is the weight vector $\beta_i \in \mathbb{R}^q$, $i = 1, \dots, C$. Thus, these three matrices are related by the following linear mapping:

$$\mathbf{D} = \beta^T \mathbf{H}, \tag{4}$$

where the matrices \mathbf{D} and \mathbf{H} are known, while the weight matrix β is not. The OLS solution of the linear system in Eq. (4) is given by the Moore-Penrose generalized inverse as follows:

$$\beta = (\mathbf{H}\mathbf{H}^T)^{-1} \mathbf{H}\mathbf{D}^T. \tag{5}$$

Eq. (5) can be split into C individual estimation equations, one for each output neuron i , being written as

$$\beta_i = (\mathbf{H}\mathbf{H}^T)^{-1} \mathbf{H}\mathbf{D}_i^T, \quad i = 1, \dots, C, \tag{6}$$

where \mathbf{D}_i denotes the i -th row of matrix \mathbf{D} .

In several real-world problems the matrix $\mathbf{H}\mathbf{H}^T$ can be singular, impairing the use of Eq. (5). In fact, a near singular $\mathbf{H}\mathbf{H}^T$ (yet invertible) matrix is also a problem, because it can lead to numerically unstable results. To avoid both problems, a common approach involves the use of the ridge regression method (a.k.a. Tikhonov regularization), which is given by

$$\beta_i = (\mathbf{H}\mathbf{H}^T + \lambda \mathbf{I})^{-1} \mathbf{H}\mathbf{D}_i^T, \quad i = 1, \dots, C, \tag{7}$$

where the constant $\lambda > 0$ is the regularization parameter.

As mentioned in the introduction, in what concerns the robustness of the ELM to outliers in classification problems, to the best of our knowledge, a comprehensive approach is still missing. Bearing this in mind, we propose the use of robust regression techniques to compute the output weight matrix, instead of the OLS approach. This approach is described in the next section.

3 Basics of M -Estimation

An important feature of OLS is that it assigns the same importance to all error samples, i.e. all errors contribute the same way to the final solution. A common approach to handle this problem consists in removing outliers from data and then try the usual least-squares fit. A more principled approach, known as *robust regression*, uses estimation methods not as sensitive to outliers as the OLS.

Huber [7] introduced the concept of M -estimation, where M stands for “maximum likelihood” type, where robustness is achieved by minimizing another function than the sum of the squared errors. Based on Huber theory, a general M -estimator applied to the i -th output neuron of the ELM classifier minimizes the following objective function:

$$J(\beta_i) = \sum_{\mu=1}^N \rho(e_{i\mu}) = \sum_{\mu=1}^N \rho(d_{i\mu} - y_{i\mu}) = \sum_{\mu=1}^N \rho(d_{i\mu} - \beta_i^T \mathbf{x}_\mu), \tag{8}$$

where the function $\rho(\cdot)$ computes the contribution of each error $e_{i\mu} = d_{i\mu} - y_{i\mu}$ to the objective function, $d_{i\mu}$ is the target value of the i -th output neuron for the μ -th input pattern \mathbf{x}_μ , and β_i is the weight vector of the i -th output neuron. The OLS is a particular M -estimator, achieved when $\rho(e_{i\mu}) = e_{i\mu}^2$. It is desirable that the function ρ possesses the following properties:

- Property 1** : $\rho(e_{i\mu}) \geq 0$.
- Property 2** : $\rho(0) = 0$.
- Property 3** : $\rho(e_{i\mu}) = \rho(-e_{i\mu})$.
- Property 4** : $\rho(e_{i\mu}) \geq \rho(e_{i'\mu})$, for $|e_{i\mu}| > |e_{i'\mu}|$.

Parameter estimation is defined by the estimating equation which is a weighted function of the objective function derivative. Let $\psi = \rho'$ to be the derivative of ρ . Differentiating ρ with respect to the estimated weight vector $\hat{\beta}_i$, we have

$$\sum_{\mu=1}^N \psi(y_{i\mu} - \hat{\beta}_i^T \mathbf{x}_\mu) \mathbf{x}_\mu^T = \mathbf{0}, \tag{9}$$

where $\mathbf{0}$ is a $(p + 1)$ -dimensional row vector of zeros. Then, defining the weight function $w(e_{i\mu}) = \psi(e_{i\mu})/e_{i\mu}$, and let $w_{i\mu} = w(e_{i\mu})$, the estimating equations are given by

$$\sum_{\mu=1}^n w_{i\mu} (y_{i\mu} - \hat{\beta}_i^T \mathbf{x}_\mu) \mathbf{x}_\mu^T = \mathbf{0}. \tag{10}$$

Thus, solving the estimating equations corresponds to solving a weighted least-squares problem, minimizing $\sum_{\mu} w_{i\mu}^2 e_{i\mu}^2$.

It is worth noting, however, that the weights depend on the residuals (i.e. estimated errors), the residuals depend upon the estimated coefficients, and the estimated coefficients depend upon the weights. As a consequence, an iterative estimation method called *iteratively reweighted least-squares* (IRLS) [2] is commonly used. The steps of the IRLS algorithm in the context of training the ELM classifier using Eq. (6) as reference are described next.

IRLS Algorithm for ELM Training

Step 1 - Provide an initial estimate $\hat{\beta}_i(0)$ using the OLS solution in Eq. (6).

Step 2 - At each iteration t , compute the residuals from the previous iteration $e_{i\mu}(t-1)$, $\mu = 1, \dots, N$, associated with the i -th output neuron, and then compute the corresponding weights $w_{i\mu}(t-1) = w[e_{i\mu}(t-1)]$.

Step 3 - Solve for new weighted-least-squares estimate of $\beta_i(t)$:

$$\hat{\beta}_i(t) = [\mathbf{HW}(t-1)\mathbf{H}^T]^{-1} \mathbf{HW}(t-1)\mathbf{D}_i^T, \quad (11)$$

where $\mathbf{W}(t-1) = \text{diag}\{w_{i\mu}(t-1)\}$ is an $N \times N$ weight matrix. Repeat Steps 2 and 3 until the convergence of the estimated coefficient vector $\hat{\beta}_i(t)$.

Several weighting functions for the M -estimators can be chosen, such as the Huber's weighting function:

$$w(e_{i\mu}) = \begin{cases} \frac{k}{|e_{i\mu}|}, & \text{if } |e_{i\mu}| > k \\ 1, & \text{otherwise.} \end{cases} \quad (12)$$

where the parameter k is a tuning constant. Smaller values of k leads to more resistance to outliers, but at the expense of lower efficiency when the errors are normally distributed. In particular, $k = 1.345\sigma$ for the Huber function, where σ is a robust estimate of the standard deviation of the errors².

In a sum, the basic idea of the proposed approach is very simple: replace the OLS estimation of the weight vector $\hat{\beta}$ of the i -th output neuron described in Eq. (6) with the one provided by the combined use of the M -estimation framework and the IRLS algorithm. From now on, we refer to the proposed approach by *Robust ELM* classifier (or ROB-ELM, for short). In the next section we present and discuss the results achieved by the ROB-ELM classifier on synthetic and real-world datasets.

4 Simulations and Discussion

As a proof of concept, in the first experiment we aim at showing the influence of outliers in the final position the decision curve between two nonlinear separable

² A usual approach is to take $\sigma = \text{MAR}/0.6745$, where MAR is the median absolute residual.

data classes. For this purpose, let us consider a synthetic two-dimensional dataset generated according to a pattern of two intertwining moons (see Fig. 1). The ELM and the ROB-ELM classifiers are trained twice. The first time they are trained with the outlier-free data set with $N = 120$ samples. The second time, they are trained with N_{out} outliers added to the original dataset. It is worth mentioning that all data samples are used for training the classifiers, since the goal is to visualize the final position of the decision line and not to compute recognition rates.

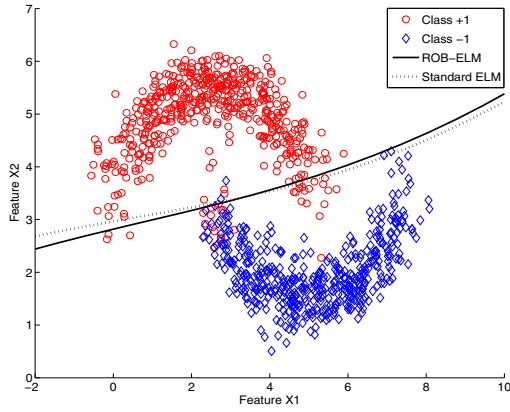
For this experiment, the Andrews weighting function was used for implementing the ROB-ELM classifier and the regularization constant required for implementing the standard ELM classifier was set to $\lambda = 10^{-2}$. Three hidden neurons with hyperbolic tangent activation functions were used for both classifiers. For the sake of fairness, the ELM and the ROB-ELM classifiers used the same input-to-hidden-layer weights, which were randomly sampled from a uniform distribution between $(-0.1, +0.1)$. The default tuning parameter k of Matlab's `robustfit` function was used. In order to evaluate the final decision curves of the ELM and the ROB-ELM classifiers in the presence of outliers, we added $N_{out} = 10$ outliers to the dataset and labelled them as belonging to class +1. The outliers were located purposefully far from the class boundary found for the outlier-free case; more specifically, at the decision region of class -1.

The results for the training without outliers are shown in Fig. 1a, where as expected the decision curves of both classifiers are similar. The results for the training with outliers are shown in Fig. 1b, where this time the decision curve of the standard ELM classifier moved (bended) towards the outliers, while the decision line of the ROB-ELM classifier remained unchanged, thus revealing the robustness of the proposed approach to outliers. The dataset (with and without outliers) used in the first experiment available by the authors upon request.

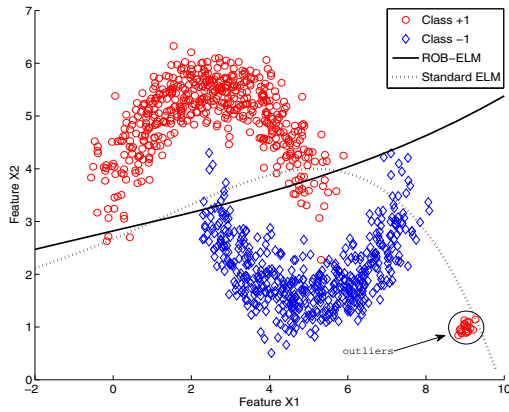
In the second and third experiments we aim at evaluating the robustness of the ROB-ELM classifier using a real-world dataset. For this experiment, four weighting functions (Bisquare, Fair, Huber and Logistic) were tested for implementing the ROB-ELM classifier and the regularization constant required for implementing the standard ELM classifier was set to $\lambda = 10^{-2}$. The default tuning parameter k of Matlab's `robustfit` function was adopted for all weighting functions.

In order to evaluate the classifier's robustness to outliers we follow the methodology introduced by Kim and Ghahramani [9]. Thus, the original labels of some data samples of a given class are deliberately changed to the label of the other class. A benchmarking dataset was chosen (Ionosphere), which is publicly available for download from the UCI Machine Learning Repository website [3].

The Ionosphere dataset describes a binary classification task where radar signals target two types of electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not, since their signals pass through the ionosphere. This dataset is comprised of 351 34-dimensional data points, and two classes (good and bad).



(a) Dataset without outliers.



(b) Dataset with outliers.

Fig. 1. Decision curves of the standard ELM and the proposed robust ELM classifiers. (a) Dataset without outliers. (b) Dataset with outliers.

We labelled the data points of class Good ($N_g = 225$ samples) and class Bad ($N_b = 126$ samples) as +1 and -1, respectively. 50% of the available samples are randomly selected for training purposes. In addition, outliers are built by randomly selecting a certain percentage P_{out} of the training samples from Class +1 and changing their labels to class -1. The testing set is outlier-free since the goal of the experiment is to evaluate the influence of outliers in the construction

Table 1. Performance comparison of the ELM and ROB-ELM classifiers for the *Iono-sphere* data set ($P_{out} = 5\%$)

q	ELM ($\lambda = 0.01$)	ROB-ELM (Bisquare)	ROB-ELM (Fair)	ROB-ELM (Huber)	ROB-ELM (Logistic)
10	70.94±4.95	69.77±6.72	72.21±4.66	72.30±4.27	72.23±4.05
15	70.67±4.53	70.76±5.54	73.41±3.34	73.11±3.40	73.56±3.75
20	70.93±3.90	72.29±5.87	74.00±3.49	74.07±3.39	74.44±3.40
25	71.24±3.33	72.89±4.86	73.94±2.94	74.51±3.35	74.41±2.87
30	71.90±3.16	72.06±4.54	75.50±2.56	75.81±2.19	75.30±2.41
35	71.81±3.01	73.90±3.49	76.36±1.75	76.13±1.51	76.27±1.62
40	72.73±2.53	73.87±3.35	76.20±2.10	76.69±1.96	76.60±2.03
45	73.34±2.25	73.46±4.05	76.69±1.83	76.83±1.86	76.93±1.89
50	73.27±2.33	73.71±4.36	77.21±2.02	76.80±2.07	76.76±2.18
100	73.97±2.03	71.49±5.42	77.03±3.68	76.46±4.32	77.03±4.17

Table 2. Performance comparison of the ELM and ROB-ELM classifiers for the *Iono-sphere* data set ($P_{out} = 10\%$)

q	ELM ($\lambda = 0.01$)	ROB-ELM (Bisquare)	ROB-ELM (Fair)	ROB-ELM (Huber)	ROB-ELM (Logistic)
10	67.03±5.12	69.77±4.80	71.06±4.52	70.57±4.77	70.47±4.03
15	66.39±5.35	70.61±4.94	70.73±4.79	68.79±5.03	70.63±4.73
20	65.57±5.71	72.04±4.77	70.86±4.06	70.56±4.76	70.90±3.98
25	65.93±4.50	73.37±4.25	72.03±3.19	71.64±3.81	71.49±4.08
30	65.83±4.07	72.17±3.69	73.17±3.09	72.40±3.48	73.04±3.08
35	66.51±3.82	73.46±3.26	73.89±2.79	73.17±2.71	73.81±2.68
40	67.17±2.78	74.56±3.01	74.86±2.94	75.07±2.78	74.60±2.90
45	66.96±3.00	75.23±3.73	75.23±3.22	74.67±2.95	75.63±2.72
50	67.27±2.57	75.33±3.66	75.67±2.79	75.66±3.14	75.97±2.74
100	68.81±2.37	72.99±6.02	77.07±4.41	76.90±4.19	77.01±4.50

Table 3. Performance comparison of the ELM and ROB-ELM classifiers for the *Iono-sphere* data set ($P_{out} = 20\%$)

q	ELM ($\lambda = 0.01$)	ROB-ELM (Bisquare)	ROB-ELM (Fair)	ROB-ELM (Huber)	ROB-ELM (Logistic)
10	50.03±5.19	50.93±5.22	55.16±6.95	50.51±5.82	53.20±6.93
15	50.07±4.49	51.43±5.53	55.20±4.86	50.79±4.64	53.87±4.99
20	48.63±3.63	52.39±4.99	55.99±6.00	50.51±4.33	54.74±5.05
25	49.83±4.32	52.31±4.25	56.46±5.64	50.90±4.41	54.11±4.49
30	49.49±3.35	53.21±3.47	56.93±4.38	51.26±4.00	55.26±3.94
35	49.90±3.22	54.76±4.39	58.71±4.90	52.84±4.37	56.69±4.62
40	49.24±2.79	57.53±5.80	61.11±6.28	56.37±5.44	59.03±6.09
45	49.93±3.03	58.41±6.83	62.76±6.92	58.69±6.73	61.04±5.87
50	50.16±3.02	61.93±6.68	64.51±7.83	60.76±7.40	63.61±6.82
100	51.43±3.18	68.94±7.89	69.91±6.97	67.03±9.39	68.39±7.67

of the decision borders of the classifiers. For this purpose, we evaluate the performances of the ELM and ROB-ELM classifiers for $P_{out} = 5\%, 10\%$ and 20% and for different values of q (number of hidden neurons). The results are given in Tables 1, 2 and 3. In these tables, we show the values of the classification rates and the corresponding standard deviations averaged over 100 training/testing runs.

By analyzing the results, we can verify firstly that the performances of all variants of the proposed ROB-ELM classifier tend to improve with an increase in the number of hidden neurons. Secondly, the performances deteriorate with an increase in the number of outliers, as expected.

As a major result one can easily verify that the performances of the ROB-ELM classifier is better than the standard ELM, specially when using the Fair, Huber and Logistic weighting functions. While the improvements in the performances of the ROB-ELM classifier are higher for higher values of q (number of hidden neurons), there are no significant improvements in the standard ELM classifier when q increases, specially for $P_{out} = 10\%$ and 20% .

As a final remark, it is worth mentioning once again that the excellent performances of the proposed ROB-ELM classifier were achieved using the default values of the tuning parameter k of Matlab's `robustfit` function for all weighting functions used in this paper. This is particularly interesting for the practitioner who wants to obtain fast and accurate results without spending much time in long fine-tuning runs of the classifier.

5 Conclusion

In this paper we introduced a robust ELM classifier (ROB-ELM) for supervised pattern classification in the presence of labeling errors (outliers) in the data. The ROB-ELM classifier was designed by means of M -estimation methods which are used to compute the weight matrix operator instead of using the ordinary least squares solution. By means of computer simulations on synthetic and real-world datasets we have shown that the resulting classifier is more robust to outliers than the standard ELM classifier.

Currently, we are further evaluating the performance of the ROB-ELM on other binary classification datasets and also on multiclass problems. The results we obtained so far suggests that this is a promising approach.

References

1. Deng, W., Zheng, Q., Chen, L.: Regularized extreme learning machine. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009, pp. 389–395 (2009)
2. Fox, J.: Applied Regression Analysis, Linear Models, and Related Methods. Sage Publications (1997)
3. Frank, A., Asuncion, A.: UCI machine learning repository (2010), <http://archive.ics.uci.edu/ml>
4. Horata, P., Chiewchanwattana, S., Sunat, K.: Robust extreme learning machine. *Neurocomputing* 102, 31–44 (2012)
5. Huang, G.B., Wang, D.H., Lan, Y.: Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics* 2, 107–122 (2011)
6. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: Theory and applications. *Neurocomputing* 70, 489–501 (2006)
7. Huber, P.J.: Robust estimation of a location parameter. *Annals of Mathematical Statistics* 35(1), 73–101 (1964)
8. Huber, P.J., Ronchetti, E.M.: Robust Statistics. John Wiley & Sons, LTD. (2009)
9. Kim, H.-C., Ghahramani, Z.: Outlier robust gaussian process classification. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) SSPR&SPR 2008. LNCS, vol. 5342, pp. 896–905. Springer, Heidelberg (2008)

10. Lee, C.C., Chiang, Y.C., Shih, C.Y., Tsai, C.L.: Noisy time series prediction using m -estimator based robust radial basis function neural networks with growing and pruning techniques. *Expert Systems and Applications* 36(3), 4717–4724 (2009)
11. Lee, C.C., Chung, P.C., Tsai, J.R., Chang, C.I.: Robust radial basis function neural networks. *IEEE Transactions on Systems, Man, and Cybernetics - Part B* 29(6), 674–685 (1999)
12. Li, D., Han, M., Wang, J.: Chaotic time series prediction based on a novel robust echo state network. *IEEE Transactions on Neural Networks and Learning Systems* 23(5), 787–799 (2012)
13. Liu, N., Wang, H.: Ensemble based extreme learning machine. *IEEE Signal Processing Letters* 17(8), 754–757 (2010)
14. Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., Lendasse, A.: OP-ELM: Optimally pruned extreme learning machine. *IEEE Transactions on Neural Networks* 21(1), 158–162 (2010)
15. Miche, Y., van Heeswijk, M., Bas, P., Simula, O., Lendasse, A.: TROP-ELM: a double-regularized ELM using LARS and Tikhonov regularization. *Neurocomputing* 74(16), 2413–2421 (2011)
16. Mohammed, A., Minhas, R., Jonathan Wu, Q.M., Sid-Ahmed, M.A.: Human face recognition based on multidimensional PCA and extreme learning machine. *Pattern Recognition* 44(10-11), 2588–2597 (2011)
17. Neumann, K., Steil, J.: Optimizing extreme learning machines via ridge regression and batch intrinsic plasticity. *Neurocomputing* 102, 23–30 (2013)
18. Zong, W., Huang, G.B.: Face recognition based on extreme learning machine. *Neurocomputing* 74(16), 2541–2551 (2011)

Handling Inconsistencies in the Revision of Probability Distributions

Fabian Schmidt¹, Jan Wendler¹, Jörg Gebhardt¹, and Rudolf Kruse²

¹ ISC Gebhardt, Celle, Germany

² Otto-von-Guericke University, Magdeburg, Germany

Abstract. In the past years knowledge-based systems gained high relevance for complex industrial applications. As storing and structuring uncertain knowledge about high-dimensional domains had become better handleable, reasoning in such domains appeared to be a more interesting field of research.

The important belief change operation called *revision* has been used to adapt probability distributions to new sets of beliefs. However, in complex real world applications *inconsistencies* are almost unavoidable when trying to assign a new set of low-dimensional (conditional) probability distributions in order to revise a high-dimensional initial distribution.

In this paper we introduce a framework for the proper handling of inconsistencies. Furthermore, we provide a method for explaining inconsistencies which occur during the revision of probability distributions. We formally introduce an algorithm to determine a minimal set of revision assignments which explains an inconsistency. Furthermore, we demonstrate how this framework has been successfully implemented in a complex application, where knowledge representation is realised with the aid of Markov networks.

1 Introduction

Due to changes in business environments, knowledge needs to be adapted to new beliefs and conditions on a regular basis. This raises the question on what should be expected from a method that performs such an adaptation automatically given the changes. In most cases only a part of the knowledge needs to be adapted. Therefore, it should be possible to specify changes locally without the need to reformulate the whole knowledge base. In case of knowledge represented as probabilities, those local specifications can be understood as conditional probabilities. Furthermore if changes are specified locally, it should be expected that only those adaptations are made to the knowledge that necessarily need to be made according to the specifications, as well as adaptations that need to be made as consequences of the specified changes. The rest of the knowledge base should remain unaffected. The latter property has been introduced to the literature as *principle of minimal change* [1].

The revision operation [2, 1, 3] addresses the need for a method which satisfies such requirements. The revision is a belief change operation that transfers

knowledge into posterior knowledge respecting the *principle of minimal change*. Furthermore, adaptations that are inferred by given changes are also applied automatically. The revision operation has been used in the context of probability distributions to adapt them to a set of given conditional probability statements. The revision computes the probability distribution which incorporates all the specified changes as well as the adaptations inferred from them to the original distribution, and is the closest distribution to the original one in the sense of the Kullback-Leibler cross entropy (see i.e.[4]), which is a well-known information-theoretical distance measure. As a consequence, the underlying probabilistic interaction structure is not altered unless required by given changes.

The probability distributions in real life application usually have a high number of dimension. For example, weather data usually consist of a huge number of variables and many of those have strong dependencies. One example for those might be temperature and the kind of precipitation. The probability of snow is much higher when the temperatures are below zero degrees. It is computational very expensive to store and operate on such high dimensional probability distributions. For that reason techniques have been developed to decompose high dimensional distributions in a number of lower dimensional distributions utilising conditional (in-)dependencies. Those methods have been introduced as probabilistic graphical models [4–7]. The most prominent of those models are Bayesian networks [8] and Markov networks [9]. Both apply graph structures in order to specify conditional probabilistic independencies, where Bayesian networks refer to directed acyclic graphs, and Markov networks to undirected graphs.

When dealing with a large network structure it becomes clear that specifying a revision problem by a large list of conditional probability statements, increases the likelihood of an inconsistent specification of the posterior probability distribution, especially when most of the statements affect different areas of the network. An inconsistent specification results in an unsolvable revision problem. It is therefore necessary to handle inconsistencies that occur in the context of the revision operation.

Recently, the problem of handling inconsistencies has been investigated for the revision of Markov networks [10]. Resolving inconsistencies automatically by so-called partition mirrors has been proposed and is now used successfully in that context [11]. Partition mirrors help to make an unsolvable revision problem solvable by slightly modifying the original revision problem.

In this contribution we present a framework for handling inconsistencies systematically using the above mentioned techniques and adding new aspects to the topic.

In Section 2 of this work the revision operation and the concept of inconsistencies are introduced. Section 3 describes the framework for handling inconsistencies in systematic manner. Section 4 illustrates methods to implement this framework using Markov networks. In Section 5 we wrap up the contribution with a summary of the results as well as some remarks for future research.

2 Fundamentals

In this section the revision operation itself and the types of inconsistencies that may occur during the revision are specified.

2.1 The Revision Operation

As mentioned before the goal of (probabilistic) revision is to compute a posterior probability distribution which satisfies the new distribution conditions, only accepting a minimal change of the quantitative interaction structures of the underlying prior distribution.

Formally spoken, in our setting a revision operation (see [2, 10]) operates on a joint probability distribution $P(V)$ on a set $V = \{X_1, \dots, X_n\}$ of variables with finite domains $\Omega(X_i)$, $i = 1, \dots, n$. The purpose of the operation is to adapt this $P(V)$ to new sets of beliefs. The beliefs are formulated in a so-called **revision structure** $\Sigma = (\sigma_s)_{s=1}^S$. This structure consists of **revision assignments** σ_s , each of which is referred to a (conditional) assignment scheme $(R_s|K_s)$ with a **context scheme** K_s , $K_s \subseteq V$, and a **revision scheme** R_s , where $\emptyset \neq R_s \subseteq V$ and $K_s \cap R_s = \emptyset$. The pair $(P(V), \Sigma)$ is called **revision problem**. For example in the revision assignment $(\text{NAV:nav1} | \text{Country:France}) = 0.2$, which would set the probability for the navigation system nav1 in the Country France to 0.2, the **context scheme** K_s would be country and the **revision scheme** R_s would be NAV.

The result of the revision, and the solution to the revision problem, is a probability distribution $P_\Sigma(V)$ which

- satisfies the revision assignments (the postulated new probabilities)
- preserves the probabilistic interaction structure as far as possible

By preserving the interaction structure we mean that, except from the modifications induced by the revision assignments σ_s all probabilistic dependencies of $P(V)$ are preserved. This requirement ensures that changes are made according to the *principle of minimal change*.

It can be proven (see, i.e. [2]) that in case of existence, the solution of the revision problem $(P(V), \Sigma)$ is uniquely defined. This solution is determined using iterative proportional fitting, which will converge to the desired distribution.

2.2 Inconsistencies

From a practical point of view it is almost impossible, even for experts, to formulate solvable revision problems. The reason for this is the fact that revision structures often contradict some of the restrictions given by zero values in the initial probability distribution $P(V)$. Note that a revision assignment may induce to change some probabilities $P(\omega) = 0$ to a strictly positive value. This kind of modification is not conform to the dependency preservation requirement of the revision operator, as zero probabilities show the absence of any interaction structure. In some contexts zero values may also refer combinations that are strictly forbidden and can therefore never be modified.

Inconsistencies have been analysed and two types of inconsistencies of revision problems have been distinguished [10]:

Inner consistency of a revision structure Σ is given, if and only if a probability distribution exists that satisfies the revision assignments of Σ ; otherwise we refer to *inner inconsistencies* of Σ .

Given that Σ has the property of inner consistency, it is still possible, that due to the zero values mentioned earlier, the revision problem $(P(V), \Sigma)$ is not solvable, since a modification of the interaction structure of $P(V)$ would be necessary. Therefore a second type of inconsistency is defined as follows: Given that Σ has the property of inner consistency, the revision problem $(P(V), \Sigma)$ shows the property of *outer inconsistency*, if and only if there is no solution to this revision problem.

Ideally *inner consistency* can be expected. However, in real world applications a large number of revision assignments as well as different interests in the parties specifying the revision assignments may still lead to an inconsistent formulation of the revision problem.

Outer inconsistencies are much harder to avoid since they rely on the probability distribution with its zero values. Those zero values cannot be changed by the revision operation.

Both types are relevant in real world applications and should therefore be addressed when attempting to handle inconsistencies in a methodical and systematic way.

3 Framework for Handling Inconsistencies

In order to approach the handling of inconsistencies in a systematic way we identified four main components namely:

- Detection
- Analysis
- Explanation / Presentation
- Automatic resolution

In the following sections the purpose and the requirements for each of those components is explained.

Detection. The first step in handling inconsistencies appropriately is detection. In order to resolve an inconsistency we first need to know that it occurred. In theory the detection of inconsistencies during the revision operation is fairly easy to achieve. The operation converges to a limit distribution if and only if no inconsistency occurred [2]. Consequently, if the revision problem Σ is not solvable, which is the case if the problem shows inconsistencies, the revision operation does not converge.

However, it is not reasonable to wait indefinitely, in order to decide whether a revision is converging very slowly or diverging between different limit distributions. The actual challenge is therefore to reliably detect the convergence of the revision operation.

Furthermore, it should be ensured that both types of inconsistencies are detected and ideally properly classified.

Analysis. If an inconsistency has been detected, it needs to be resolved in order to create a solvable revision problem $(P(V), \Sigma)$. Resolving inconsistencies manually by an expert using his domain knowledge is often times preferable to an automatic resolution by an algorithm. However, to experts inconsistencies are most of the time not obvious. For that reason an automated analysis is necessary in order to help experts to manually resolve inconsistencies according to their domain knowledge.

This component therefore should offer one or more methods to analyse inconsistencies in an automated way. Even though *inner inconsistencies* are relatively easy to understand once they are spotted, both types of inconsistencies should be addressed.

Usually the root cause of an inconsistency is of interest. However, if multiple inconsistencies occur it is not easy to identify what causes an inconsistency initially. For that reason the analysis might focus on one particular inconsistency and try to identify all components that together compose this inconsistency. The result should be a collection of information or clues that explain the given inconsistency.

Explanation / Presentation. After the analysis, the results need to be presented to a human expert. Raw results of an analysis are often not easy to understand by humans. It is therefore useful to process the results, or resulting clues, of an analysis and automatically generate an explanation in order to help an expert understand and appropriately resolve inconsistencies.

Ideally such an explanation visualises the inconsistency in an appropriate way. However, what is appropriate often times depends on the individual expert. For that reason it is helpful to offer different kind of explanations to suit different needs of experts. Graphical visualisations as well as a textual explanations seem reasonable.

Automatic Resolution. Although manual resolution of inconsistencies is preferable, in practical application it is not always feasible to resolve each and every occurring inconsistency manually. Therefore also a mechanism to resolve inconsistencies automatically is needed. Such mechanism should ensure that the revision operation always returns a distribution that is consistent to the (modified) revision assignments.

Some reasonable requirements for such a method are that after the resolution the revision structure Σ has the property of *inner consistency*, and Σ is also consistent to the underlying interaction structure so *outer consistency* is given. Furthermore, the adaptation should be made in a way that the resulting limit distribution is as close to the originally specified distribution as possible. There should be no adaptation unnecessary to resolve the inconsistency.

4 Implementing the Framework for Markov Networks

After defining the requirements for the framework, this section outlines how it is used with the revision operation for Markov networks. We proceed in a different order of the components because for some of them good solutions have been proposed already. We first describe detection and automatic resolution and then address the aspects that have not yet been solved.

Detection. The revision operation as it is implemented is embedded in a revision control mechanism [10]. This method detects inconsistencies. Since the revision operation is an iterative operation the number of iterations has been limited. Either convergence is achieved before this limit is reached or the revision stops after this number of iterations and starts to use partition mirrors to resolve an inconsistency. Convergence will be determined using a heuristic method. Convergence will be assumed when the revision error becomes smaller than a set threshold or when the change compared to the previous iteration is smaller than a given minimal value.

This method does detect inconsistencies well. However sometimes the algorithm detects convergence because the changes between two iterations are very small or even too small to detect given the computational accuracy we have. That can have two effects. Either a specific inconsistency is not detected although in reality an inconsistency occurred, because convergence was detected. Or an inconsistency gets detected as the convergence is not yet reached, although with enough iterations the problem would be solvable.

Automatic Resolution. To resolve inconsistencies during the revision of Markov networks automatically the method of partition mirrors [11] has proven itself to be suitable. In principle it extends the interaction structure by introducing new mirror variables and then uses the revision operation with all its benefits to resolve the inconsistencies. This method results in a modified revision structure Σ_{rev} that then shows *inner consistency* and furthermore forms a revision problem with the property of *outer consistency*. This approach has the elegant property that it modifies the revision assignments with respect to the *principle of minimal change*. This leads to a distribution that is closest to one specified by the inconsistent original revision structure Σ .

Analysis. Analysing the root cause of an inconsistency is not a trivial endeavour. One theoretically good way to analyse inconsistencies by observing probability mass flows called epsilon revision has been described in [10]. The idea behind this revision is to replace zero values in the probability distribution that indicate the absence of an interactive structure by very small values ϵ . After that, given *inner consistency*, the revision problem is always solvable. By observing former zero values that now get significant probability mass, inconsistencies become visible. While this approach works great in theory it is computationally very expensive in practical application. For that reason a heuristic method is presented here.

The algorithm we present was developed under the assumption, that an automatic resolution took place, modifying the revision structure using partition mirrors. It operates using the modified revision structure Σ_{rev} , the probability distribution P_{rev} (in this case a Markov Network) which is the result of the automatic resolution and the original revision assignment σ^* for which an inconsistency should be analysed. Essentially, such an analysis is used to find out which inconsistency has caused the modification of a certain revision assignment σ^* during the automatic resolution.

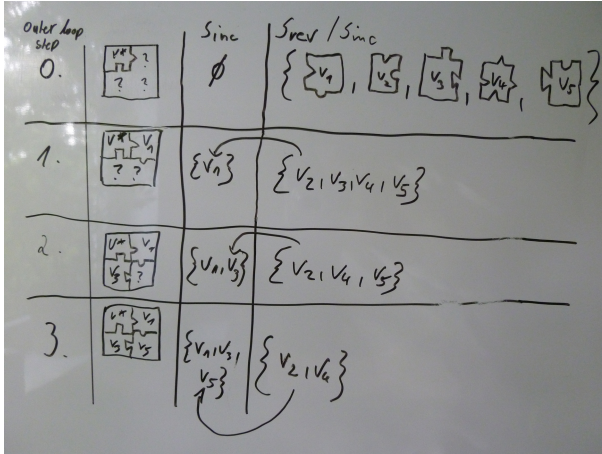


Fig. 1. Algorithm used for finding puzzle pieces: outer loop collects pieces, inner loop finds fitting piece

Algorithm to Determine a Minimal Explaining Set of Revision Assignments. For the description of the algorithm, the revision structure Σ will be represented by a set S that contains all the revision assignments $\sigma_i \in \Sigma$. Furthermore, $revision(P, S)$ is donated to a revision operation performed on the revision problem (P, Σ) . The revision operation returns an error value. If it converges to a single limit distribution, the error is zero. If it does not converge, it will stop after a certain number of iterations and return an error value which can be used further. The method to compute the error value should have the following properties:

- $error = 0$, if the revision operation converges
- $error(P, S) \leq error(P, S \cup \sigma)$
- it should be easily computable

As such error function the Shannon Entropy (see i.e. [4]) could be used, but it is computationally expensive. For that reason in practical applications preferably heuristic methods are used.

To make the algorithm easier to understand, the set S_{rev} represents the revision structure Σ_{rev} that was obtained by the automatic resolution. It contains all the revision assignments that together formed the revision structure except σ^* which is to be analysed. P_{rev} is donated to the probability distribution obtained after the original revision. S_{inc} represents the minimal explaining set and is iteratively filled during the algorithm. S_{test} is another set of revision assignments that is used to determine the revision assignments that will be included into S_{inc} .

Giving the parameters as specified above, the algorithm performs its search as follows:

```

maxError = revision( $P_{rev}, S_{rev} \cup \sigma^*$ )
 $S_{inc} := \emptyset$ 
repeat
   $S_{test} := \emptyset; i := 0$ 
  repeat
     $i := i + 1$ 
     $S_{test} := S_{test} \cup \{\sigma_i\}$   $\sigma_i \in S_{rev} \setminus S_{inc}$ 
     $e_i = revision(P_{rev}, S_{test} \cup \sigma^* \cup S_{inc})$ 
    if  $e_i \geq \text{maxError}$  then
       $S_{inc} := S_{inc} \cup \sigma_i$ 
  until  $e_i \geq \text{maxError}$ 
   $e_{inc} = revision(P_{rev}, S_{inc} \cup \{\sigma^*\})$ 
until  $e_{inc} \geq \text{maxError}$ 

```

In the outer loop every step identifies one new revision assignment σ_i which contributes to the inconsistency. It is moved from S_{rev} to S_{inc} which represents the minimal explaining set. This is schematically shown in figure 1. If this set is sufficient to explain the inconsistency completely, the algorithm ends. Otherwise it continues. Note that the algorithm ends finally if S_{inc} contains every revision assignment from S_{rev} . In that case all revision assignments are necessary to explain the inconsistency.

The inner loop identifies one revision assignments σ_i that is part of the inconsistency. Starting with an empty set S_{test} , in every iteration a new revision assignment σ_i is added to S_{test} . This set is then united with σ^* and the set S_{inc} , which contains all revision assignments contributing to the inconsistency, that have been identified so far. With this new set a revision operation $revision(P_{rev}, S_{inc} \cup \{\sigma^*\} \cup S_{test})$ is performed. If the error obtained by this revision operation reaches maxError , the last revision assignment σ_i added to S_{test} , definitely contributes to the inconsistency. Now the inner loop finishes and the outer loop can go one step further.

The algorithm always leads to a solution. It always terminates because in every step of the outer loop, S_{inc} is increased by an σ_i which was not in S_{inc} before. In the worst case S_{inc} is the initial S_{rev} , which is always a solution. In theory multiple minimal sets might exist, and the algorithm returns one of them. By applying all different orderings of the σ_i , all minimal sets can be determined by introducing an additional outer loop.

Explanation / Presentation. Currently we are using the minimal explaining set determined using the method just outlined also as explanation. This method is already used successfully in our productive environment. However, the productive use of this method showed the need to further process the minimal set. Depending on the expertise of the user the minimal set alone is not always sufficient to fully understand the inconsistency and its cause. Consequently, more elaborated methods to automatically produce more useful explanation are currently researched.

5 Conclusion

Knowledge collected by businesses today grows more and more complex. Furthermore, the environment that businesses operate in changes rapidly on a daily basis. To cope with this, methods to store and revise knowledge were developed. One popular method for storing knowledge is by representing it in form of probability distributions and using the revision operation to incorporate new beliefs while keeping changes to the original knowledge as small as possible. As this technique also allows more complex problems be addressed, inconsistencies during the revision process are almost unavoidable. In order to keep the quality of the revision process high it is important to handle inconsistencies properly.

In this work we introduced a framework for the systematic handling of inconsistencies during the revision operation. The framework has been validated using previously proposed techniques and new solutions for aspects that have not been addressed yet have been provided. We motivated the necessity to analyse inconsistencies and provided an algorithm that determines a minimal set of revision assignments that cause an inconsistency. Furthermore, this set was used as explanation for the inconsistency.

The approach is used productively at the Volkswagen Group in the system EPL which calculates part demands. World wide over a hundred planners are using this system to calculate the demands for more than a hundred different model groups from different car manufacturers within the Volkswagen Group. Before the introduction of the algorithm inconsistencies important to the planners had to be analysed manually by experts. With a few hundred inconsistencies in complicated model groups it is infeasible to analyse all inconsistencies manually. After the introduction of the automated analysis the users can now run those analysis directly from the software. This greatly reduced the manual effort for analysis experts. However there are still about five requests for further manual analysis and explanation every week. In those cases the minimal explaining set of revision assignments is not enough for the user to understand an inconsistency sufficiently. Also for data analysis experts it only offers a starting point for further manual analysis and explanation. One limitation of this approach is the number of revision assignments in the minimal set, too many revision assignments make results hard to understand. Therefore possibilities to reduce the number of revision assignments presented to a user could prove useful. One approach to achieve that is to group similar revision assignments. Another idea is to explain only parts of the inconsistency.

Furthermore, we observed that without the knowledge of the underlying dependency structure, it is hard to understand the relations between the revision assignments contained in the minimal set. It therefore seems helpful to include areas of the dependency structure in the explanation, so that more information about the problem becomes visible.

Another area of future research is the method for analysis itself. All previously mentioned points assume that a minimal set of revision assignment is computed. Other approaches like the analysis and visualisation of probability mass flow during the revision process might yield better and more understandable explanations.

References

1. Gärdenfors, P.: *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press (1988)
2. Gebhardt, J., Borgelt, C., Kruse, R., Detmer, H.: Knowledge revision in markov networks. *Mathware & Soft Computing* 11(2-3), 93–107 (2004)
3. Gabbay, D.M., Smets, P. (eds.): *Handbook of Defeasible Reasoning and Uncertainty Management Systems. Belief Change*, vol. 3. Kluwer Academic Press, Netherlands (1998)
4. Whittaker, J.: *Graphical Models in Applied Multivariate Statistics*. Wiley (1990)
5. Lauritzen, S.L.: *Graphical Models*. Oxford Statistical Science Series. Oxford University Press, USA (1996)
6. Borgelt, C., Steinbrecher, M., Kruse, R.: *Graphical Models - Representations for Learning, Reasoning and Data Mining*. Wiley (2009)
7. Kruse, R., Schwecke, E., Heinsohn, J.: *Uncertainty and Vagueness in Knowledge Based Systems: Numerical Methods*. Springer, Berlin (1991)
8. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann (1991)
9. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B* 50(2), 157–224 (1988)
10. Gebhardt, J., Klose, A., Wendler, J.: Markov network revision: On the handling of inconsistencies. In: Moewes, C., Nürnberger, A. (eds.) *Computational Intelligence in Intelligent Data Analysis. SCI*, vol. 445, pp. 153–165. Springer, Heidelberg (2013)
11. Klose, A., Wendler, J., Gebhardt, J., Detmer, H.: Resolution of inconsistent revision problems in markov networks. In: Kruse, R., Berthold, M., Moewes, C., Gil, M.A., Grzegorzewski, P., Hryniewicz, O. (eds.) *Synergies of Soft Computing and Statistics. AISC*, vol. 190, pp. 517–524. Springer, Heidelberg (2013)

Creating Knowledge Base from Automatically Extracted Information*

Beata Nachyla

Institute of Computer Science, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
B.Nachyla@ii.pw.edu.pl

Abstract. In this article we present a self-learning method for discovering the domain specific knowledge contained in a set of text documents. The method assumes that contents of the input documents have tagged domain-relevant information. The information is tagged with labels from a prespecified set. The method counts the co-occurrences of various sequences of the labels in a sentence and represents them in form of a data structure called a Prefix Label Tree. In order to extract knowledge from a given document, we use a hierarchical clustering method to group the labels contained within the document's content. In order to calculate similarity of clusters during the clustering process, we also propose a measure called the Relation Possibility (*RP*).

1 Introduction

Automatic creation of a Knowledge Base is one of the most important tasks in the area of Text Mining research. Huge amount of related information is hidden in text resources taking up even terabytes of memory and distributed over the Internet. No one is able to control and maintain it manually. We need tools for automatic extraction of information pieces from the resources and discovering relationships connecting them. A user needs to specify what is the knowledge he is interested in. The part of knowledge defined by the user is a domain specific knowledge since it depends on a selected domain of interest. To get benefit of the automatically gathered knowledge we also need tools for browsing, visualizing and modifying the knowledge. Ontologies and Knowledge Bases are graph-based methods of knowledge representation and specification, very convenient to visualize and browse. In this paper, we present a self-learning method for discovering a domain specific knowledge contained in a set of text documents. The method assumes that the contents of input documents have tagged domain-relevant information. The information is tagged with labels from a prespecified set. The offered learning method counts the co-occurrences of various sequences of the labels in sentences and represents them in a form of a data structure called a Prefix Label Tree. In order

* This work was supported by the National Centre for Research and Development(NCBiR) under Grant No. SP/I/1/77065/10 devoted to the Strategic scientific research and experimental development program: 'Interdisciplinary System for Interactive Scientific and Scientific-Technical Information'.

to extract knowledge from a given document, we use a hierarchical clustering to group the labels contained within the document's content. In order to calculate similarity of clusters during the clustering process, we also offer a measure called Relation Possibility (*RP*). Having the clusters of labels, we look for the knowledge patterns inside each cluster. The *knowledge patterns* are meaningful units of the knowledge that must be specified by user. Each pattern defines a meaningful part of the domain knowledge being contained in the document. Our research is carried out as the part of the SYNAT project [7]. The project concerns creation of a universal platform for gathering and hosting scientific information and knowledge. Most of the information is available in the Internet and has form of unstructured text resources. We had to develop methods for automatic information extraction from such resources and in turn to discover the knowledge contained in their contents. The scientific domain knowledge for the project purposes is presented in form of *SYNAT System Ontology of Science* [6]. Beneath, we state the problem of the knowledge discovering we had to solve as a part of the project. We had given set of text resources containing the information tagged according to methods we previously prepared [2]. Our task was to propose methods for deriving a knowledge base constituted of instances of concepts and relations defined in the SYNAT System Ontology representing the knowledge contained in the resources. The most general version of the problem assumes that the set of text resources consists of documents that have some specified type, e.g. we might have a set of persons' or academic institutions' homepages. The method should be general and independent of the type of the documents and their structure. We assume that a set of labels used for tagging information in the documents is available. As we assume that no training data shows existing correspondences between the information tags and concepts and relation from the ontology, we require our method to be unsupervised.

The state-of-the-art in automatic knowledge discovery can be separated into the following three main areas: 1) extraction of an ontology and knowledge base during a same process (see e.g. [9]), 2) discovery of groups of information pieces that are related, but without any knowledge definition, like ontologies (see e.g. [3]), 3) discovery of instances of the objects defined by a knowledge specification that is already given (see e.g. [1]). The research we present in this paper is related to third area. We assumed that a domain ontology is already given and knowledge patterns are specified by a user. Our method derives the instances of that patterns from a set of unstructured texts. There are works that apply approaches similar to ours, e.g. the aim of the method proposed in [1] is to discover instances of given knowledge patterns. The authors made assumption similar to ours that information units appearing in a same sentence or paragraph are related. In contrast to our approach, their method always creates an instance of a pattern if tags forming the pattern occur in a same sentence. On the other side, if semantically related information tags are distributed over several paragraphs the corresponding instance of the pattern will never be discovered. Such a situation can appear in case of web pages, where the related information can be presented in several lines because of graphical structure of the page, e.g. in order

to emphasise information about person's work a position name is presented in one line and a related affiliation is given below, in the next line. In our method, we do not rely only on a structure of a document. We use the information about co-occurrences' counts of tags in a same sentence to build a model. In the next step we select the instances of possible patterns which are the most probable according to that model. We still rely on an assumption that related information pieces are adjacent in text, but we gain an independence on a structure of a document. The method we propose is more flexible and resistant on grammatical and structural form of the text.

Our paper has the following layout. In Section 2, we give a preliminary notions concerning Text Mining. We define concepts used in further considerations. Section 3 contains a detailed description of the method proposed in the article. Section 4 presents experiments performed according to this method. Section 5 contains a summary and directions for future work.

2 Preliminaries on Text Mining, Ontologies, and Knowledge Base Building

The aim of Text Mining is an exploration of information existing in unstructured text resources by means of algorithmic methods. A text mining methodology consists of two phases [8]:

- a *Text Refining*, i.e. converting a free-form text into an Intermediate Form (IF) (see: [8]), which in turn is a structured input for the second phase,
- a *Knowledge Distillation* i.e. extracting useful information patterns and relations between them from the intermediate form of the text.

In this paper, a *domain specific knowledge* is defined as a set of concepts and relationships (relations) among them. A convenient and popular way of representing the domain specific knowledge is an ontology. The *ontology* is a graph, where the nodes correspond to the concepts and the edges between them describe the relationships. (see to [4] for details).

The *information extraction* is usually a process of searching for fragments of text that describe relevant information (specified by the domain). The selected fragments of text are tagged with *labels* (called also the *tags*) from a specified set by this process. The set definition is a part of the domain-relevant information specification, which has to be prepared by the user. Each label corresponds to some *information unit* (e.g. name of a person or a location, list of ingredients of a food product). The process of inserting the labels into document's content is called *tagging*. The text of a document with the labels placed within is called a *tagged text* (a *tagged document*). The relevant knowledge extracted from text consists of objects being instances of the concepts, and instances of the relations defined in the domain ontology. Hence the extracted knowledge is a graph of the instances of relevant nodes and edges from the ontology. The graph of instances is called a *Knowledge Base*.

```

Steve received his [bsdegree] B.S. [/bsdegree] in [bsmajor]Computer Science [/
bsmajor] from [bsuniv]National Taiwan University [/bsuniv] in [bsdate]1991[/
bsdate], and his [msdegree] M.S. [/msdegree] and [phddegree] Ph.D. [/phddegree
] in [phdmajor]Electrical Engineering [/phdmajor] from [phduniv]Stanford
University [/phduniv] in [msdate]1995[/msdate] and [phddate]2000[/phddate],
respectively. While at Intel, he has conducted research in developing new
program analyses and programming environments to exploit advanced
microarchitectures. Prior to Intel, he worked for 8 years on the Stanford SUIF
compiler project (part of National Compiler Infrastructure) which delivered
the highest SPEC FP number at the time of the 1998 paper listed below. His
thesis work is an interactive interprocedural parallelizer called the SUIF
Explorer. (...) He is a [position]member [/position] of [affiliation]ACM [/
affiliation] and [affiliation]IEEE [/affiliation].

```

Fig. 1. Example document tagged in information extraction process

Here we want to emphasise the difference between the information patterns and the knowledge patterns. The information extraction process uses natural language processing and statistical techniques to discover syntactic patterns called *information patterns*. The ontology describes semantic concepts and relations. The knowledge base building process is an extraction of semantic patterns, called *knowledge patterns* in this article. Beneath, we illustrate the introduced notions.

Example 1. Let us consider a short text document presented in Fig. 1. The document is a part of a homepage, taken from the dataset used for testing information extraction methods, which is available at [5]. The document's content is enriched with the labels being a result of tagging information of interest during an information extraction process. A domain of our interest is a career of people working in science. We want to retrieve information and knowledge about one's work positions in respective affiliating organizations and gained science degrees on three educational levels: a Bachelor of Science (BSc), a Master of Science (MSc) and a Doctorate (PhD). The following set of labels was used to tag the domain relevant information:

- *position, affiliation* - two labels used to tag employment information,
- *bsdegree, msdegree, phddegree* – labels used to tag a name of gained science degree: BSc, MSc and PhD, respectively,
- *bsuniv, msuniv, phduniv* – labels used to tag the name of an academic organization, where a respective degree was obtained,
- *bsdate, msdate, phddate* – labels used to tag the date of a final diploma exam,
- *bsmajor, msmajor, phdmajor* – labels used to tag the name of a science discipline, which was the subject of a diploma thesis.

The set of labels can be split into four groups with respect to four information units forming information patterns (see Section 2):

- $\{position, affiliation\}$ - unit regarding an employment information,
- $\{bsdegree, bsuniv, bsdate, bsmajor\}$ – unit regarding a BSc degree graduation,
- $\{msdegree, msuniv, msdate, msmajor\}$ – unit regarding an MSc graduation,
- $\{phddegree, phduniv, phddate, phdmajor\}$ – unit regarding a PhD graduation.

Fig.2 presents an ontology defining the domain knowledge. In fact, the ontology is a part of the Synat Ontology of Science [6], with some simplifications. Fig.4 shows the part of knowledge discovered from the example document. The information regarding MSc and PhD graduations was omitted to make presented graph more readable. We might indicate three instances of sample knowledge units in this graph. The first one covers instances of concepts and relations regarding a Steve’s membership in ACM. The second covers the instances regarding Steve’s membership in IEEE. The last one covers a subgraph concerning a BSc graduation. The first and the second are instances of a same knowledge unit, that we call an employment pattern. Let us notice that the pattern corresponds to the information unit regarding an employment. The third one is an instance of a knowledge unit named a degree pattern. Degree pattern corresponds to three information units regarding a BSc, an MSc and a PhD graduation. Employment pattern and degree pattern are examples of knowledge patterns, i.e. the

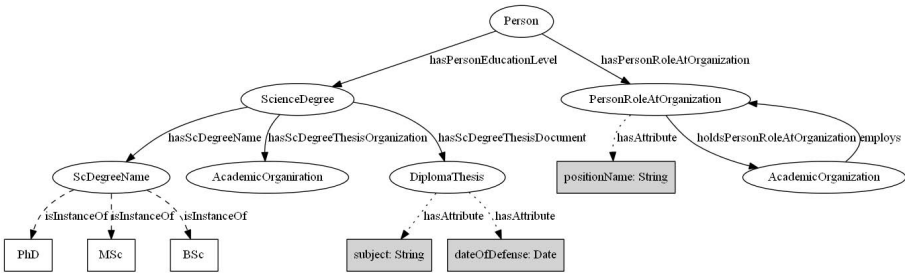


Fig. 2. Example ontology graph

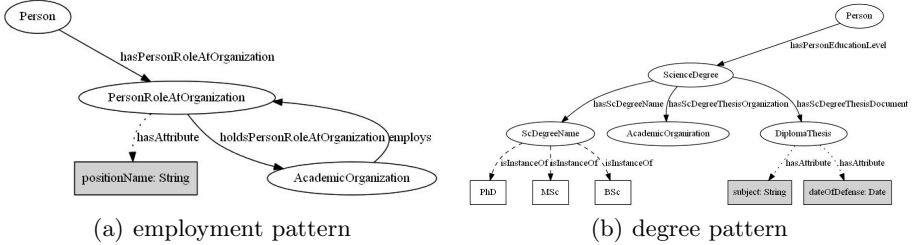


Fig. 3. Knowledge patterns regarding knowledge about employment and graduation

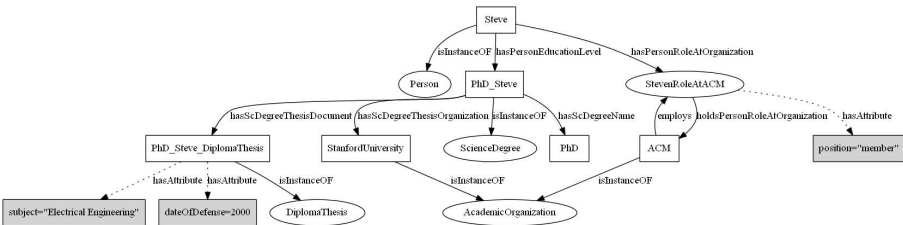


Fig. 4. A knowledge extracted from the document in Fig. 1 in a graph form

subgraphs of ontology graph covering meaningful for user parts of knowledge. Fig. 1 shows subgraphs presenting the employment pattern and the degree pattern.

3 New Method of Building a Knowledge Base

In this section, we will offer a new method of building a knowledge base from automatically extracted information from text documents.

Let C be a set of texts previously processed by an information extraction process. Each document in C has domain relevant information tagged with labels from a prespecified set. An example of such a document is presented in Fig. 1. The main idea of the proposed method is based on counting co-occurrences of the labels in the same sentence. We make an assumption, that labels occurring often in a sentence are very likely related, whereas an existence of relation between labels that rarely appear in a same sentence is not as frequent. A Prefix Label Tree structure is built in order to represent the counts of co-occurrences in the sentences taken from documents in C . Having the tree and a document with relevant information tagged (the document has to be tagged with the labels appearing in the tree), we derive a set of instances (i.e. relations and concepts from the domain ontology). The instances should be added to the Knowledge Base in order to represent the knowledge contained in the document. The derivation is made by firstly, clustering of the labels occurring in the document, and then by matching the knowledge patterns in the obtained clusters.

The process of building the tree is described in Subsection 3.1. In Subsection 3.2 we describe a Relation Possibility (RP) measure. The hierarchical clustering of the labels, which uses the measure, is presented in Subsection 3.3.

3.1 Building a Prefix Label Tree

A prefix label tree will be built under the following two assumptions:

- labels, that often occur in the same sentence are supposed to be related,
- ordering of the labels within a sentence is insignificant (which, by the way, allows us to reduce a search space of our problem).

We start the process of building a prefix label tree with a tree consisting of a dummy node, named a root. It represents an empty labels' tuple. Having given sufficiently large set of documents C previously tagged during an information extraction process, we consider a sentence-based corpus, i.e. a set of sentences coming from all the documents in C . We count co-occurrences of each pair of the labels in all sentences. According to the second assumption, labels ordering is insignificant, so we consider only the pairs ordered lexicographically. Having counts for each pair, we continue with counting occurrences of labels triples, quadruples, and so on. Same as for pairs, the labels in longer tuples are ordered lexicographically. Each tuple is added as a new branch to the prefix tree, or makes an extension of an existing branch. The nodes of the outcoming tree correspond

to the labels, whereas the branches represent the tuples of labels. Each edge with a source node s and a target node t has a corresponding number. It shows a co-occurrence count of the pair (s, t) , in a certain sentence, which also contains all the labels from preceding nodes.

3.2 A Relation Possibility Measure

In this section, we propose a new method of measuring a possibility of semantic relation existence between the labels forming a sequence. Let us consider a new document d processed by the information extraction process. Let $S = (l_1, l_2, \dots, l_n)$ be a sequence of all labels contained in d in order as they appear in the text. A same label can occur on many positions in the S sequence, but if the positions are adjacent the label is placed in the sequence only once.

For an arbitrary k -element subsequence S' of sequence S one may want to calculate a possibility that fragments of the text marked with the labels are related semantically and form a knowledge pattern. Here we propose a method of the possibility estimation that uses the co-occurrence counts present in a label prefix tree. To use the tree we need to order S' lexicographically. Let us denote that ordered tuple by $S_{ord} = (t_1, t_2, \dots, t_k)$. We search for all the paths in the tree corresponding to S_{ord} . Assume that we have the following found m paths corresponding to S_{ord} in the tree: (p^1, p^2, \dots, p^m) . For each of the paths we are interested in the occurrence counts that describe edges connecting respective nodes. Hence, let us assume that a path p^i is a sequence of numbers $c^i(t_j)$ being occurrences of the labels' pairs: $p^i = (c^i(t_1), c^i(t_2), \dots, c^i(t_k))$, $i = 1, \dots, m$. We define an occurrence count of a path $c(p^i)$ as the minimal occurrence count of its elements:

$$c(p^i) = \min_{j=1..k} \{c^i(t_j)\}. \quad (1)$$

We define an occurrence count of an ordered sequence of labels $c(S_{ord})$ as the maximal occurrence count of the paths in the labelled prefix tree corresponding to the sequence:

$$c(S_{ord}) = \max_{i=1..m} \{c(p^i)\}. \quad (2)$$

Finally, we define a measure of relation existence possibility between the labels forming a sequence S_{ord} as:

$$RP(S_{ord}) = \frac{c(S_{ord})}{d(S_{ord})}, \quad (3)$$

where $d(S_{ord})$ is a normalizing coefficient being the maximum of maximal occurrences of elements in the paths: $d(S_{ord}) = \max_{i=1..m} \{\max_{j=1..k} \{c^i(t_j)\}\}$.

3.3 Hierarchical Clustering of Labels

In this section, we propose a method of discovering knowledge patterns' instances from information patterns previously tagged. We make an assumption, that related information units appear in a text as a sequence of adjacent labels. Is it a heuristic assumption - not always true, but fulfilled in the most of the cases. Hence we perform a hierarchical clustering of adjacent labels.

Let us consider an n -element sequence of all labels appearing in a given document d , defined as in Eq. 3.2. We start the clustering process with a set of n one-element clusters: $\{\{l_1\}, \{l_2\}, \dots, \{l_n\}\}$. Having the set of values of relation possibility measure for each pair of adjacent labels, we select a pair with maximal value of that measure (let us denote this pair by (l_i, l_{i+1})). The clusters containing the pair are merged. New set of clusters we receive is as follows: $\{\{l_1\}, \{l_2\}, \dots, \{l_i, l_{i+1}\}, \dots, \{l_n\}\}$. The set of values of relation possibility measure is updated by two operations. First, by removing the values corresponding to the following pairs (l_{i-1}, l_i) , (l_i, l_{i+1}) , (l_{i+1}, l_{i+2}) and then by inserting new values calculated for the following tuples: (l_{i-1}, l_i, l_{i+1}) , (l_i, l_{i+1}, l_{i+2}) . Again, the maximal value of the measure is selected and corresponding tuples are merged. The process continues until the maximal value of the measure is lower than a specified threshold. The threshold is a parameter of the clustering algorithm and has to be selected by the user. We end up with a set of groups of adjacent labels. For each group of labels in the set, we select information patterns, which can be formed from the labels. The instances of knowledge patterns, that correspond to the information patterns, are created and inserted into a knowledge base.

4 Experiments

In this section we describe two experiments we performed. We used documents from [5] for testing purpose. The documents were tagged with over 20 labels, but we selected 11 of them. The reduction of the label set let us keep the results of the experiments more readable and easier to explain. The selected labels are the same as the labels used in Example 1. Remaining labels were skipped during processes of building Prefix Label Tree and labels' clustering. The first experiment was as follows. In the first step, a prefix label tree was built from a set of all documents available in [5]. Then for each document the labels' hierarchical clustering was performed, as described in 3.3. A clustering measure threshold was set to 0.0 hence clusters were being merged until one cluster was obtained. Sample processes of clustering are illustrated in Fig. 5 and 6. A sample results of the first experiment are showed in Fig. 5(a) and 6(a). Diagrams in these figures show clustering process in the form of a tree. The leaves correspond to one-element initial clusters i.e. labels contained in a document. A leaf contains the name of the corresponding label and the tagged text. An internal node corresponds to a new cluster, that is the result of merging the clusters corresponding to nodes which are its predecessors.

In the second experiment we used, so called, generalized labels. We split the 11 labels (the same as in Example 1) into four following groups constituting more general information units:

- *bsdate*, *msdate* and *phddate* we call, more generally, as a *date*,
- *affiliation*, *bsuniv*, *msuniv* and *phduniv* we call as a *organization*,
- *bsmajor*, *msmajor* and *phdmajor* we call, more generally, as a *discipline*,
- *position* remains the same.

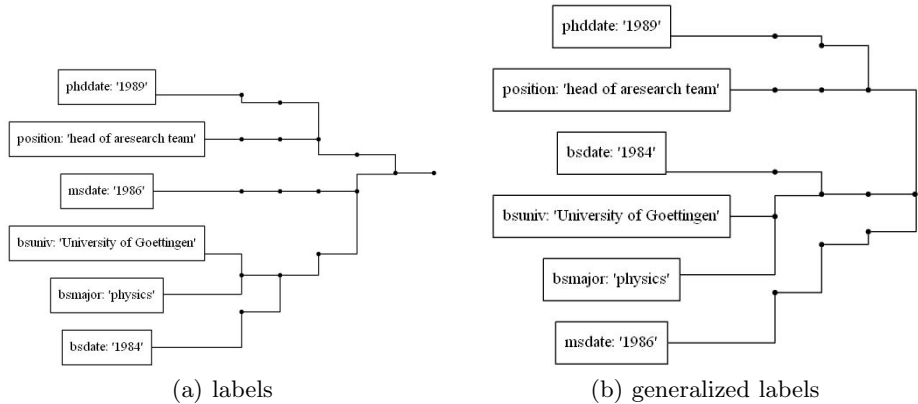


Fig. 5. Results of experiments performed on document *38016.txt* from [5]

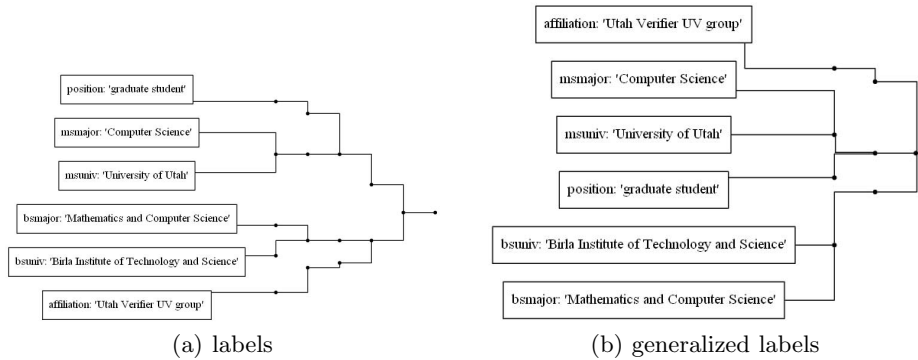


Fig. 6. Results of experiments performed on document *38110.txt* from [5]

As in the previous experiment a prefix label tree was built. The labels appearing in all the documents were changed to generalized labels and used to build the tree. Then, for each document labels contained in it, they were changed to generalized labels and clustered. Sample results of the experiment in the form of clustering trees are presented in Fig. 5(b) and 6(b). The clustering trees' leaves contain the original labels, not generalized ones, since 11 selected labels correspond to information units (see 2). We performed the second experiment to test our method for its ability to extract knowledge patterns' instances from partial or more general information about information units. The example knowledge patterns we presented in Section 2 are, in some sense, simple to derive. Each information unit is comprised by only one knowledge pattern, hence there is a one to one correspondence between the information units (labels) and the knowledge patterns. Hence, one might consider deriving instances of knowledge patterns by splitting labels contained in a document into groups according to the corresponding knowledge patterns. The result might be similar to the results obtained in the first experiment. By generalizing the original labels we showed that our method is able correctly to extract many (but not all) instances of

knowledge patterns even if there is a non-trivial correspondence between them and information units.

Since we have no training data for the problem of deriving instances of knowledge patterns, we enclose clustering trees for 130 documents from [5] in the following archive <http://www.ii.pw.edu.pl/~bnachyla/results.zip>. We hope it might be helpful in rating the accuracy of our method.

5 Conclusions and Future Work

In this article, we presented new method of discovering knowledge from unstructured texts. We used Prefix Label Tree to concisely represent information about co-occurrence counts of label sequences. Then we proposed clustering of labels contained in a document to derive instances of knowledge patterns. A new measure (Relation Possibility) of clusters' similarity were provided. The whole method is unsupervised and can extract knowledge from a same set of documents, which were used to build a co-occurrence counts model. Each label in instance of knowledge pattern discovered according to our method occurs not more than once. In our future work we plan to extend our method to allow searching for the instances of knowledge patterns in which a same label can appear many times. Another research task we plan to carry out is testing of similarity measure influence on obtained results.

References

1. Alani, H., Kim, S., Millard, D.E., Weal, M.J., Hall, W., Lewis, P.H., Shadbolt, N.: Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems* 18(1), 14–21 (2003)
2. Andruszkiewicz, P., Nachyła, B.: Automatic extraction of profiles from web pages. In: Bembenik, R., Skonieczny, Ł., Rybiński, H., Kryszkiewicz, M., Niezgódka, M. (eds.) *Intelligent Tools for Building a Scientific Information. SCI*, vol. 467, pp. 415–432. Springer, Heidelberg (2013)
3. Culotta, A.: Dependency tree kernels for relation extraction. In: *Proc. of ACL 2004*, pp. 423–429 (2004)
4. Guarino, N., Oberle, D., Staab, S.: What is an ontology? In: Staab, S., Studer, R. (eds.) *Handbook on Ontologies*, 2nd edn. Springer (2009)
5. <http://arnetminer.org/lab-datasets/profiling/>
6. Synat system ontology, <http://wizar.ii.pw.edu.pl/passim-ontology/>
7. <http://www.synat.pl/>
8. Tan, A.H.: Text mining: The state of the art and the challenges. In: *Proc. of PAKDD 1999*, pp. 65–70 (1999)
9. Xiao, C., Zheng, D., Yang, Y., Shao, G.: Automatic domain-ontology relation extraction from semi-structured texts. In: Zhang, M., Li, H., Lua, K.T., Dong, M. (eds.) *IALP*, pp. 211–216. IEEE Computer Society (2009)

A HMM-Based Location Prediction Framework with Location Recognizer Combining k -Nearest Neighbor and Multiple Decision Trees

Yong-Joong Kim and Sung-Bae Cho

Department of Computer Science, Yonsei University
50 Yonsei-ro, Seodaemun-gu, Seoul 120-749, Korea
{yongjoong, sbcho}@yonsei.ac.kr

Abstract. Knowing user's current or next location is very important task for context-aware services in mobile environment. Many researchers have tried to predict user location using their own methods. However, they focused mainly the performance of method, and only few were considered development of real working system on mobile devices. In this paper, we present a location prediction framework, and develop a personalized destination prediction system based on this framework using smartphone. The framework consists of two methods of recognizing user location based on the combined method of k -nearest neighbor (k NN) and decision tree, and predicting user destination based on the hidden Markov model (HMM). The destination prediction system is composed of four parts including mobile sensor log collector, location recognition module, location prediction module, and system management module. Experiments on real datasets of five persons showed that our method achieved average prediction accuracy above 87%.

Keywords: Location recognition, Location prediction, Location extraction, Location-based services.

1 Introduction

With the ubiquity and ever-increasing capabilities of mobile devices, smartphone has become a powerful platform to be exploited for mobile context-aware services. Moreover, because a variety of sensors have been equipped in recent mobile devices, we could get much information from the sensors. In this regard mobile context-aware services have attracted more attention, and active investigation about inferring user's mobile contexts is actively being conducted for the services [5, 12, 16, 17]. Above all one of the most important user contexts is location. It allows information and services in the mobile device to be localized. It means the proper services and information can be delivered according to user's current location or future location. In spite of a lot of research with respect to location prediction, however, the real working system which recognizes and predicts the location on the mobile device has not been developed is still on the way.

In this paper, we propose a location prediction framework, and develop the personalized location prediction system for mobile context-aware services. The contributions of this paper can be summarized as follows.

- Development of the real working system: we develop a location prediction system by integrating location recognition module, location prediction module, sensor log collector, and system management module.
- Exploitation of G-means algorithm: for user trajectories to be discriminatively modeled we extract the intermediate locations by using G-means clustering method which determines the number of clusters automatically by performing statistical test iteratively.
- Management of models and user data: we develop user interface for learning recognition and prediction model and implement the functions of location management and path management for the system to appropriately manage user's locations and paths.

The rest of this paper is organized as follows. Section 2 briefly reviews the related works. Section 3 presents the details of the proposed framework. Section 4 presents the personalized location prediction system. Section 5 conducts some experiments and analyzes the experimental results. Section 6 concludes the work.

2 Related Works

Ashbrook et al. extracted user's significant locations from GPS data and presented a location predictor based on the Markov models [2]. Krumm et al. designed a method called predestination that predicts driver's destination as trip progresses [9]. Alvarez-Garcia et al. presented a new approach to predict destinations given only data of a partial trip by using hidden Markov models (HMMs) and local street-map [1]. Simmons et al. proposed a HMM-based approach by using a map database and GPS sensor to providing real-time predictions on driver destination and route [23]. Mathew et al. designed a hybrid method for predicting human mobility on the basis of HMM [13]. Petzold et al. presented a dynamic Bayesian network to predict an indoor next location and compared with the state predictor and multi-layer perceptron predictor [18]. Yavas et al. presented a data mining algorithm for the prediction of user movements in a mobile computing system [24]. Monreale et al. proposed trajectory pattern tree aimed at predicting with a certain level of accuracy the next location of a moving object [14]. Morzy mined the database of moving object locations to discover frequent trajectories and movement rules, and matched the trajectory of a moving object with the database of movement rules to build a probabilistic model [15].

Petzold et al. compared various methods for next location prediction [19]. In their work, the comparing experiments were conducted using dynamic Bayesian network, multi-layer perceptron, Elman net, Markov predictor and state predictor. Scellato et al. presented a novel framework for predicting user's next locations based on nonlinear time series analysis [22].

3 Location Prediction Framework

3.1 Problem Definition

Basically, the location prediction problem is as follows. Given an observed user trajectory $T = \ell_0 \rightarrow \ell_1 \rightarrow \ell_2 \rightarrow \dots \rightarrow \ell_t$ where ℓ_i is the i th location user visited and the set of locations, $L = \{L_1, L_2, L_3, \dots, L_n\}$; predict ℓ_{t+1} , i.e., the next location of the user. Fig. 1 shows the problem as a pictorial representation.

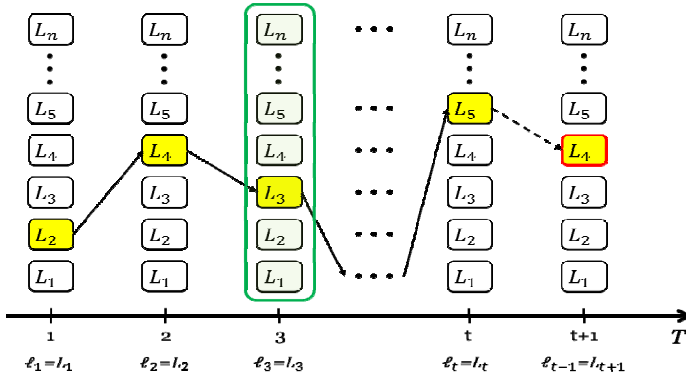


Fig. 1. Fundamental location prediction problem

In this paper, to solve the location prediction problem we transform the problem into the path classification problem as follows. Given an observed user trajectory, $T = \ell_0 \rightarrow \ell_1 \rightarrow \ell_2 \rightarrow \dots \rightarrow \ell_t$, the set of locations $L = \{L_1, L_2, L_3, \dots, L_n\}$, the set of user’s movement paths $P = \{p_1, p_2, p_3, \dots, p_m\}$ where $p_i = (\ell_s^i, \ell_d^i)$ (subscript s and d denote start and destination indicators, respectively), and the set of intermediate locations $I = \{I_1, I_2, I_3, \dots, I_n\}$ between start location and destination location of each path p_i ; classify an observed user trajectory T into one of movement paths of user. By solving this problem, we can get a classified path and predict user’s next location by returning destination location of the path. Fig. 2 shows that the location prediction problem is converted to the path classification problem.

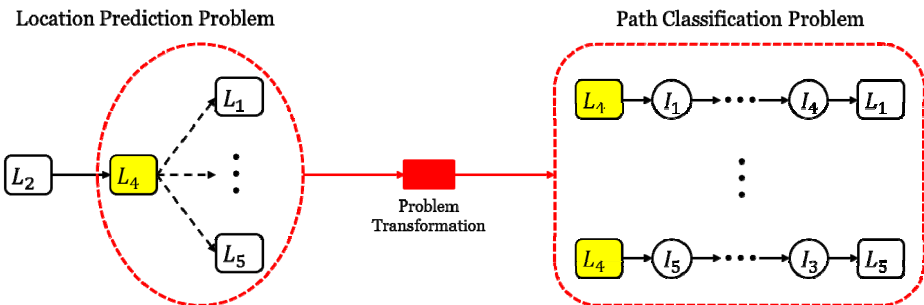


Fig. 2. Problem transformation

3.2 The Proposed Framework

The proposed framework for predicting user’s next location is briefly described as a graphical representation in Fig. 3. In this figure, we introduce a plate (the box labeled t) that represents t nodes of which three nodes are shown explicitly and a red rectangle to denote the problem transformation. We also introduce extra nodes such as T_n and S_n , which mean the time when user visits the n th location and transportation mode when user visits the n th location, respectively. The information of these nodes enables user’s path to be modeled discriminatively.

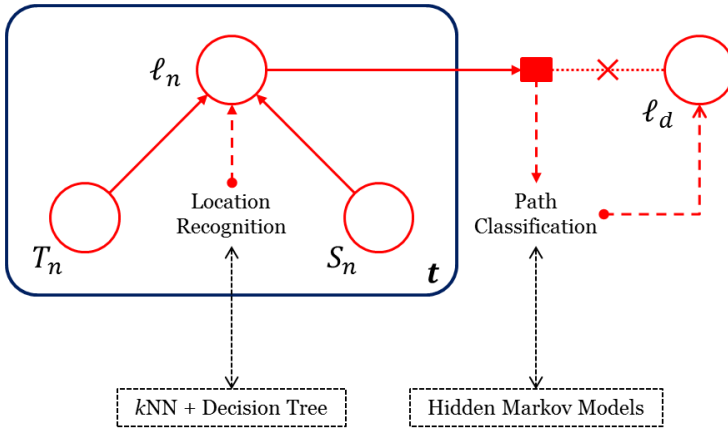


Fig. 3. Graphical representation of the proposed framework

Location Recognition. Location recognition task is very important for predicting user location because sequence of locations is used to learn the path classification model. The green box in Fig. 1 represents the location recognition task. For recognizing user’s current location, we use the decision tree based method integrated with k NN. A decision tree is a tree-like graph that uses a branching method to illustrate every possible outcome of a decision, and it could operate efficiently in mobile devices because it can be trained in fast time [4, 11].

Algorithm 1. Location Recognition

Input : Position (Current GPS coordinate)
Output : Identified location name or “Unknown”

- 1: $N \leftarrow kNN(\text{Position})$ /* candidate locations */
- 2: **for** $k = 1, \dots, N$ **do**
- 3: $\text{result} \leftarrow \text{DecisionTree}(k)$
- 4: **if** ($\text{result} = \text{“Yes”}$) **then**
- 5: **return** $\text{GetLocationName}(k)$
- 6: **end if**
- 7: **end for**
- 8: **return** “Unknown”

The location recognition method firstly performs k NN at user's current position, and then filters the locations of which current position is out of their boundary. Secondly, decision trees of the filtered locations are invoked to identify current user location. As the input of the method, we use two types of information, GPS and Received Signal Strength Indicator (RSSI) of Wi-Fi access point. Algorithm 1 shows the location recognition method as a pseudocode.

Location Extraction. To model user path discriminatively, we extract the intermediate locations between start location and destination location by clustering GPS data. Most of previous works based on clustering method used k -means clustering method [2, 7, 8]. The k -means clustering method, however, is not suitable for real-world system because it needs the pre-knowledge about the number of k . Instead of using k -means clustering algorithm, we use Gaussian-means method, which is simply called G-means clustering method. The method is based on statistical test for the hypothesis that a subset of data follows a Gaussian distribution, and automatically chooses the number of clusters k by iteratively performing k -means clustering method until the test accepts the hypothesis that the data assigned to each k -means center are Gaussian [6].

Algorithm 2. Location Prediction

Input : Observation sequence

$$O_{seq}: o_1 \rightarrow o_2 \rightarrow o_3 \rightarrow \dots \rightarrow o_m$$

Output : Predicted destination \hat{D}

```

1: Path  $\leftarrow$  GetPaths( $o_1$ ) /* user paths starting from  $o_1$ 
   */
2: result := 0
3: index := 0 /* for path index */
4: for k = 1, ..., |Path| do
5:   temp  $\leftarrow$  HMMPath(k)( $O_{seq}$ )
6:   /* finding the index having maximum value */
7:   if (result < temp) then
8:     result := temp
9:     index := k
10:  end if
11: end for
12:  $\hat{D} \leftarrow$  GetDestinationOfPath(index)
13: return Predicted location  $\hat{D}$ 

```

Path Classification. For classifying user path, we use HMMs as a classifier. A HMM is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. HMMs are known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition and so on [21]. The use of HMMs for path classification enables us to account for location characteristics as hidden states, and also to account for the effects of each individual's previous actions.

Equation (1) is the probability of the observation sequence O , given the HMM parameter λ_i of user path p_i . We use multiple information to make observation symbol of HMMs. The symbol is generated by using location information from location recognition, the time when user visits location, and transportation mode classified by exploiting accelerometer, magnetic and orientation sensors [7].

$$P(O|\lambda_i) = \sum_Q P(O|Q, \lambda_i)P(Q|\lambda_i) \tag{1}$$

Equation (2) is the HMM classifier.

$$\hat{P} := \operatorname{argmax}_i P(T|\lambda_i) \tag{2}$$

We describe the location prediction algorithm as a pseudocode in Algorithm 2.

4 Personalized Location Prediction System

In this section, we present a personalized location prediction system based on our framework. The system is composed of the sensor log collector, location recognition module, destination prediction module, and system management module. Fig. 4 shows the architecture of the location prediction system.

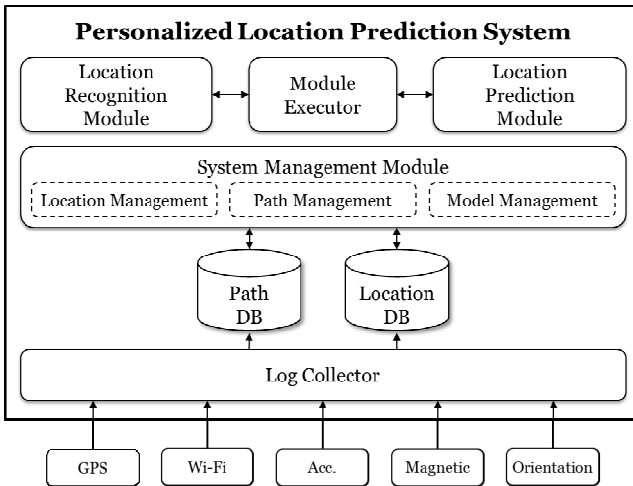


Fig. 4. Architecture of the personalized location prediction system

4.1 Log Collector

We implemented mobile sensor log collector, and integrated it into our system. The types of sensors that we collected are shown in Table 1, and we use collection period as 0.5 second. Fig. 6(a) shows the implemented log collector.

Table 1. Various types of sensors used for data collection

No.	Sensor	Description
1	GPS	Latitude, Longitude
2	Wi-Fi	MAC address, RSSI
3	Acceleration	3-axis double type data
4	Magnetic Field	3-axis double type data
5	Orientation	Orientation, Pitch, Roll
6	Time Stamp	Date, Time
7	Transportation Mode	Staying, Walking, Vehicle, Subway

4.2 Location Recognition and Prediction Module

Fig. 5 shows the flow of location prediction. Our framework for predicting user’s next location is implemented in the location recognition and prediction modules.

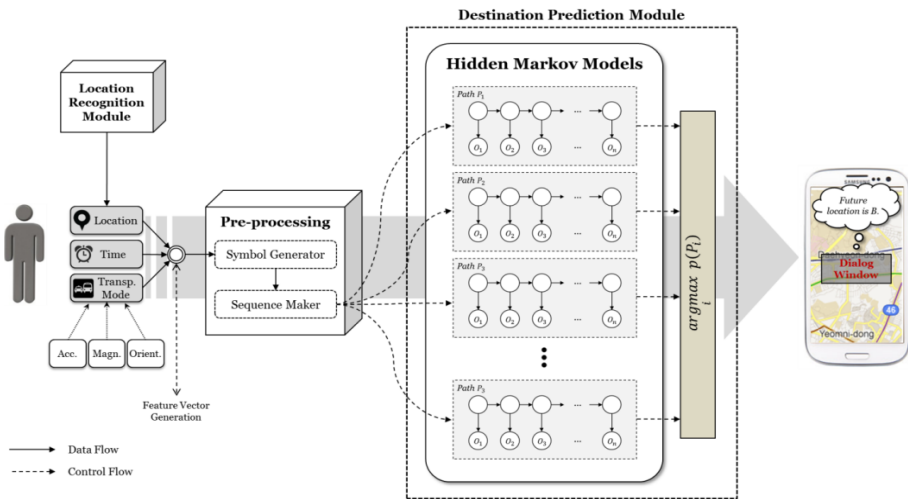


Fig. 5. Flow of location prediction

4.3 System Management Module

System management module consists of three parts including location management, path management, and model management. In location management, the locations labeled by user are managed in relational database. The user can add new locations, modify the information of registered locations, and delete existing locations. Fig. 6(c) shows a screenshot of location setting. The paths of user are also managed in relational database. Model management is responsible for learning location recognition and prediction models. Fig. 6(d) is an example of learning location prediction model in smartphone.

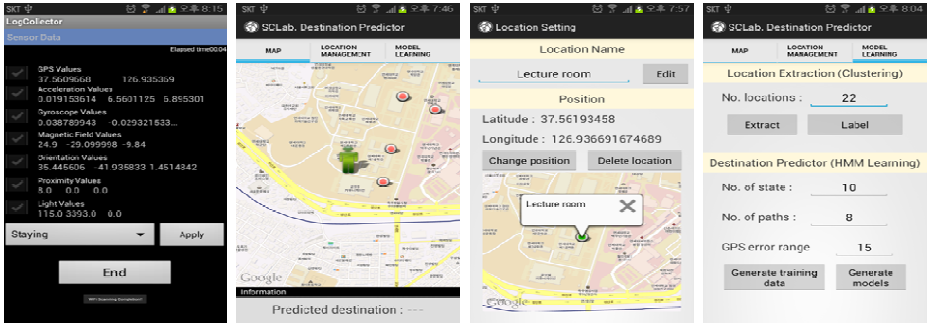


Fig. 6. Screenshots of the implemented system; (a) mobile sensor log collector; (b) initial map screen; (c) specific location setting; (d) model learning for destination prediction

5 Experiments

To show the performance of our prediction method, we conduct an experiment for evaluating prediction accuracy. In this section, we give an account of the dataset and performance analysis in the experiments.

5.1 Dataset Description

For the experiments, five undergraduate students of Yonsei university collected smartphone sensor data presented in Table 1 for two months from September to October in 2012. We use the Samsung Galaxy S3 as the experiment device. The amount of collected data is shown in Table 2.

Table 2. The information of collected data

User	#Location	#Path	#Intermediate Location	Data Size
User1	19	15	97	514MB
User2	26	8	127	505MB
User3	31	22	202	754MB
User4	23	10	109	701MB
User5	18	7	74	246MB

5.2 Performance Analysis

To implement the prediction model, we cluster the dataset in Table 2 using G-means clustering. The number of extracted intermediate locations is shown in Table 2.

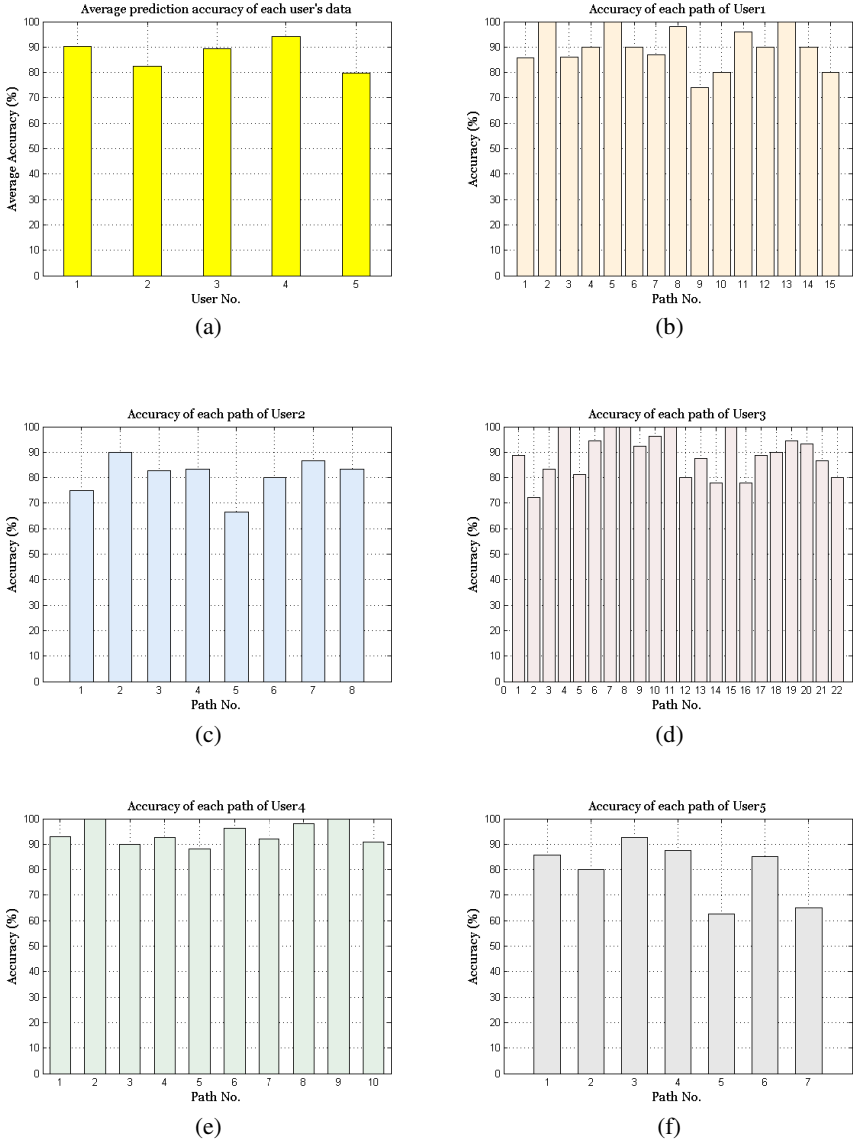


Fig. 7. Experimental results; (a) average prediction accuracy of users; (b)-(f) prediction accuracies for each user-specific movement paths

Using implemented models, we perform the leave-one-out cross-validation (LOOCV) for each dataset of the users in Table 2. Fig. 7 shows the experimental results. Fig. 7(a) shows average accuracy of each user’s dataset, and the graphs from (b) to (f) show the accuracy in respect of each path of each user, respectively.

As a result of the experiment, the accuracy of several paths of each user is less than 70%, even if the accuracy of most of paths is higher than 80%. It is because the amount of data to learn the HMMs of these paths is not enough, and the movement patterns of each path are too diverse to model the mobility of each path.

6 Conclusion

We have designed a framework for the location prediction problem, and implemented a real system working on mobile devices. Location recognition module performs the k NN to select candidate locations using GPS, and uses decision trees using RSSI of Wi-Fi access points. For location prediction, we extract the intermediate locations of the paths by using G-means clustering method, and generate HMMs using user's multiple contexts, and predict user's location using them. An experiment has been performed to evaluate the accuracy of prediction model on mobile devices. We achieved the average prediction accuracy which is higher than 87% through the experiment. As a future work, we will study about the incremental learning algorithm of location prediction model for real system to be adaptively learned through real-time data.

Acknowledgments. This research was supported by Samsung Electronics Co., Ltd.

References

1. Alvarez-Garcia, J.A., Ortega, J.A., Gonzalez-Abril, L., Velasco, F.: Trip Destination Prediction Based on Past GPS Log Using a Hidden Markov Model. *Expert Systems with Applications* 37(12), 8166–8171 (2010)
2. Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing* 7(5), 275–286 (2003)
3. Calabrese, F., Lorenzo, G.D., Ratti, C.: Human Mobility Prediction based on Individual and Collective Geographical Preferences. In: *Proceedings of the 13th IEEE Intelligent Transportation Systems*, pp. 312–317 (2010)
4. Caruana, R., Niculescu-Mizil, A.: An Empirical Comparison of Supervised Learning Algorithms. In: *Proceedings of the 23rd International Conference on Machine Learning*, pp. 161–168 (2006)
5. Chen, G., Kotz, D.: A Survey of Context-Aware Computing Research. Technical Report TR2000-381, Dartmouth (November 2000)
6. Hamerly, G., Elkan, C.: Learning the k in k -means. In: *Advanced in Neural Information Processing Systems*, vol. 16 (2003)
7. Hightower, J., Consolvo, S., LaMarca, A., Smith, I., Hughes, J.: Learning and Recognizing the Places We Go. In: Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H. (eds.) *UbiComp 2005. LNCS*, vol. 3660, pp. 159–176. Springer, Heidelberg (2005)
8. Kang, J.H., Welbourne, W., Stewart, B., Borriello, G.: Extracting Places from Traces of Locations. In: *Proceedings of the 2nd International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, pp. 110–118 (2005)

9. Krumm, J., Horvitz, E.: Predestination: Inferring destinations from partial trajectories. In: Dourish, P., Friday, A. (eds.) *UbiComp 2006*. LNCS, vol. 4206, pp. 243–260. Springer, Heidelberg (2006)
10. Lee, Y.S., Cho, S.B.: An Efficient Energy Management System for Android Phone Using Bayesian Networks. In: *Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops*, pp. 102–107 (2012)
11. Lim, T.S., Loh, W.Y., Shih, Y.S.: A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. *Machine Learning* 40(3), 203–228 (2000)
12. Löwe, R., Mandl, P., Weber, M.: Context Directory: A Context-Aware Service for Mobile Context-Aware Computing Applications by the Example of Google Android. In: *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops 2012*, pp. 76–81 (2012)
13. Mathew, W., Raposo, R., Martins, B.: Predicting future locations with hidden Markov models. In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 911–918 (2012)
14. Monreale, A., Pinelli, F., Trasarti, R.: WhereNext: a Location Predictor on Trajectory Pattern Mining. In: *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining*, pp. 637–646 (2009)
15. Morzy, M.: Mining Frequent Trajectories of Moving Objects for Location Prediction. In: Perner, P. (ed.) *MLDM 2007*. LNCS (LNAI), vol. 4571, pp. 667–680. Springer, Heidelberg (2007)
16. Noh, H.Y., Lee, J.H., Oh, S.W., Hwang, K.S., Cho, S.B.: Exploiting Indoor Location and Mobile Information for Context-Awareness Service. *Information Processing and Management* 48(1), 1–12 (2012)
17. Park, H.S., Oh, K., Cho, S.B.: Bayesian Network-Based High-Level Context Recognition for Mobile Context Sharing in Cyber-Physical System. *International Journal of Distributed Sensor Networks* (2011)
18. Petzold, J., Pietzowski, A., Bagci, F., Trumler, W., Ungerer, T.: Prediction of Indoor Movements Using Bayesian Networks. In: Strang, T., Linnhoff-Popien, C. (eds.) *LoCA 2005*. LNCS, vol. 3479, pp. 211–222. Springer, Heidelberg (2005)
19. Petzold, J., Bagci, F., Trumler, W., Ungerer, T.: Comparison of Different Methods for Next Location Prediction. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) *Euro-Par 2006*. LNCS, vol. 4128, pp. 909–918. Springer, Heidelberg (2006)
20. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc., San Francisco (1993)
21. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE* 77(2), 257–286 (1989)
22. Scellato, S., Musolesi, M., Mascolo, C., Latora, V., Campbell, A.T.: NextPlace: A spatio-temporal prediction framework for pervasive systems. In: Lyons, K., Hightower, J., Huang, E.M. (eds.) *Pervasive 2011*. LNCS, vol. 6696, pp. 152–169. Springer, Heidelberg (2011)
23. Simmons, R., Browning, B., Zhang, Y., Sadekar, V.: Learning to Predict Driver Route and Destination Intent. In: *Proceedings of the IEEE Intelligent Transportation Systems Conference 2006*, pp. 127–132 (2006)
24. Yavas, G., Katsaros, D., Ulusoy, Ö., Manolopoulos, Y.: A data mining approach for location prediction in mobile environments. *Data & Knowledge Engineering* 54(2), 121–146 (2005)

Noisy Data Set Identification

Luís Paulo F. García¹, André C.P.L.F. de Carvalho¹, and Ana C. Lorena²

¹ Computer Science Department
Institute of Mathematics and Computer Sciences, University of São Paulo
São Carlos - SP, Brazil

{lpgarcia, andre}@icmc.usp.br

² Institute of Science and Technology, Federal University of São Paulo
São José dos Campos - SP, Brazil
aclorena@unifesp.br

Abstract. Real data are often corrupted by noise, which can be provenient from errors in data collection, storage and processing. The presence of noise hampers the induction of Machine Learning models from data, which can have their predictive or descriptive performance impaired, while also making the training time longer. Moreover, these models can be overly complex in order to accomodate such errors. Thus, the identification and reduction of noise in a data set may benefit the learning process. In this paper, we thereby investigate the use of data complexity measures to identify the presence of noise in a data set. This identification can support the decision regarding the need of the application of noise redution techniques.

Keywords: Noisy data, Noise identification, Data Complexity Measures.

1 Introduction

In order to induce classification models, supervised Machine Learning (ML) techniques are applied to a labeled data set composed of n pairs (\mathbf{x}_i, y_i) , where each \mathbf{x}_i is a tuple of predictive attributes describing a certain object and y_i , named target attribute, corresponds to its class. The predictive performance of the induced model for new data depends on various factors, such as the training data quality and the inductive bias of the classification algorithm. Nonetheless, despite of the algorithm bias, when data quality is low, the performance of the predictive model is impaired.

In real world applications, there are many components that affect data quality, such as data source, the sampling period and how the information is collected [1]. Data acquisition is inherently leaned to errors even though extreme efforts are made to avoid them. Some studies estimate that even in controlled environments there are at least 5% of errors in a data set[2, 3].

Although many ML techniques have internal mechanisms to deal with noise (such as the pruning mechanism in Decision Trees [4, 5]), the presence of noise in data may lead to difficulties in the induction of ML models. These difficulties include the increase in processing time, a higher complexity of the induced model

and a possible deterioration of its predictive ability for new data [6]. When these models are used in critical environments, they may have security and reliability problems [7].

In order to reduce the problems due to the presence of noise, many authors treat them in a pre-processing step, also known as data cleaning. It normally involves using one or more filters which try to identify the noisy data. Afterwards, the identified inconsistencies can be corrected or, more often, eliminated [8]. Several studies investigate techniques for noise detection and removal, like [9–11, 8, 12–14].

In this paper we explore how features extracted from noisy and noiseless data can be used to characterize the presence or absence of noise in a data set. These attributes, extracted from the data set, measure the complexity of the classification problem [15]. For such, they consider the overlap between classes imposed by features and the separability and distribution of the data points. Experimental results show that the addition of noise in a data set affects the geometry of the classes separation, which can be captured by these measures. Using a base composed of noiseless data sets and data sets with artificial noise, classifiers able to detect the presence of noise are induced. The induced models can be used to decide whether a new data set needs to be cleaned by a noise reduction technique.

A recent work that uses complexity measures in a noise-filtering scenario is [16]. The authors employ these measures to predict whether a filtering technique is effective for cleaning a data set that will be used for the induction of nearest neighbor classifiers. This approach differs from the approach proposed here in several aspects. One of the main differences is that while the approach proposed by [16] is restricted to nearest neighbor classifiers, our approach is classifier independent.

The paper is organized as follows. Section 2 provides a definition of noise as regarded in this paper. Section 3 presents the complexity measures used to describe the problems. Section 4 discusses about the experiments carried out and the methodology applied. In Section 5 we present the data sets used and the experimental results. Finally, Section 6 summarizes the conclusions and discusses future works.

2 Noise

Several definitions of noise can be found in the ML and Statistics literature. Most of them agree that noisy data may harm the learning process, since they present inaccuracies [5]. In many studies, outliers are also regarded as noisy data, although they are actually extreme or exceptional, but correct, cases. In [17], for instance, some interesting noises and outliers in a medical domain were detected by an *ensemble* of noise detection techniques.

For supervised learning data sets, Zhu and Wu [18] distinguish two types of noise: in predictive attributes and in the target attribute. The later occurs in the class label and can be caused by errors, subjectivity in labeling or even by an

inadequate labeling process. In another study, Wu [2] argue that errors in predictive attributes tend to be systematic, as a result, for example, of uncalibrated measures.

The automatic identification of noise is a difficult task. Ideally, it should involve a validation step, where the objects highlighted as noisy are confirmed as such, before they can be further processed. Since the most common approach is to eliminate noisy data, it is important to properly distinguish these data from the safe data. Safe data need to be preserved, once they have features that represent part of the knowledge necessary for the induction of an adequate model.

In a real application, evaluating whether a given example is noisy or not usually has to rely on the judgment of a domain specialist, which is not always available. Furthermore, the need to consult a specialist tends to increase the cost and duration of the pre-processing step. This problem is reduced when artificial data sets are used, or when simulated noise is added to a data set in a controlled way. The systematic addition of noise simplifies the validation of noise detection techniques and the study of the noise influence in the learning process.

The possible approaches for imputing noise in a classification data set depend on the noise type, as described next:

1. **Target attribute Noise:** this approach usually changes the class label of some examples. There are two methods to add noise to the class label: (1) random, in which each example has the same probability of having its label corrupted (changed to another label) [12]; and (2) pairwise, in which a percentage $x\%$ of the majority class examples have their labels modified to the same label as the second majority class [13].
2. **Predictive attribute Noise:** usually there is an assumption that the correlation between the predictive attributes is weak. Thus, adding noise into a predictive attribute does not influence other predictive attributes. The noise addition is in general random and may occur with a percentage $x\%$ for a given predictive attribute, respecting its maximum and minimum values allowed [14]. Besides, a percentage $y\%$ of the predictive attributes can be corrupted.

When these types of noise are added to a data set, its examples are corrupted within a given rate. In most of the related studies noise is added according to rates ranging from 5% until 40% with intervals of 5% [18], although other papers opt for fixed rates (as 2%, 5% and 10%) [17]. For random methods, normally this addition is repeated a number of times for each noise level.

3 Data Set Complexity Measures

In order to discriminate between noiseless from noisy data sets, it is necessary describe the main features of each data set. In this study, we investigate the use of data complexity measures for such. These complexity measures are related to either topological or geometrical characteristics of the data set objects. As a

result, each data set is represented by a feature set, where each feature value is extracted by a complexity measure. And, each example represents a noiseless or a noisy data set, using different levels of noise. The target attribute label is either the presence or the absence of noise in the data set. The objective is to learn a model able to classify a new data set as noiseless or noisy.

The data complexity measures try to assess how difficult the classification problem associated with the data set is. They were initially proposed for two-class problems [15] and some were later extended to multiclass problems [19]. For the two-class measures that were not extended, we first decomposed the multiclass problem into a series of two-class problems, using the *one-vs-all* decomposition. Thus, for a data set with k classes, k binary data sets are generated, where one of the k classes is labelled as positive, while the others are negative. The average of the measure values calculated for each binary data set is then used. For distance calculations, we employed an hybrid euclidean-overlap distance measure.

There are three main groups of complexity measures, from which 13 different measures were extracted to represent the main features of the noiseless and noisy data sets [15, 19]:

- **Overlap of values from different classes.** These measures analyse the attribute values to assess the separability of the classes in a data set. The discriminant power of each attribute for each class describes its degree of ambiguity in relation to the other classes. For such, Basu et al. [15] and [19] proposed the use of the maximum Fisher’s discriminant ratio (F1), the overlap of the per-class bounding boxes (F2), the maximum individual feature efficiency (F3), the directional-vector maximum Fisher’s discriminant ratio (F1v) and the collective feature efficiency (F4).
- **Class separability.** These measures evaluate the complexity of the decision border which is necessary to discriminate the classes. For such, they analyze the separability boundaries between classes based on characteristics related to linear hyperplans and distances between examples. These measures include the minimum of an error function for a linear classifier (L1), the training error of a linear classifier (L2), the fraction of points in the class boundary (N1), the ratio of average intra/inter class nearest neighbor distance (N2) and the *leave-one-out* error rate for the one-nearest neighbor classifier (N3).
- **Measures of geometry, topology, and density.** With geometric bias, these measures use definitions as manifolds to describe the margin of separation between classes. Besides, they provide density and geometric indicators. These measures include the nonlinearity of a linear classifier (L3), the nonlinearity of the one-nearest neighbor classifier (N4) and the fraction of maximum covering spheres (T1).

4 Methodology

The experiments performed have two phases. The first phase, the pre-processing, creates noisy versions of the original data sets by using the two systematic models

of noise described in Section 2. While the class label noise was added randomly, the percentage of noisy predictive attributes varied from 10% to 40%, with intervals of 10%. These attributes were chosen according to their information gain regarding the class label, preferring attributes with higher information gain. For each noise percentage, 10 new versions of noisy data sets were produced. The complexity measures were extracted from the original data sets and from their corrupted counterparts. We used the Data Complexity Library (DCoL) [19] to calculate these measures.

As a result of the pre-processing, an unbalanced data set with 42 (#data sets) examples labeled as original, representing the noiseless versions of the data sets, and $42 (\# \text{ data sets}) * 4 (\# \text{ noise levels}) * 10 (\# \text{ random executions})$, totalizing 1680 examples labeled as noisy, was created. This data set will be named noise identification data set.

In the second phase, we investigate if we can correctly identify a original or noisy data set using the complexity measure values extracted from the data set. For such, we compared the predictive performance of five ML techniques in this task. The ML techniques used in this study are Naive Bayes (NB) [20], Random Forests (RF) [21], CART decision tree induction algorithm (CART) [22] k -Nearest Neighbor (k -NN), with $k=3$ [23] and Support Vector Machines (SVM) [24].

5 Experimental Results

This section describes the main aspects of the experimental results performed in this study.

5.1 Data Sets

We selected 42 labeled data sets from the UCI repository [25] for the experiments, keeping a low number of class labelling inconsistencies. Although some of these data sets were artificially generated, most of them represent real problems. Table 1 summarizes the main characteristics of these data sets: number of examples, number of attributes, number of classes and percentage of the examples in the majority class.

For each data set, random noise levels were inserted, with rates of 5%, 10%, 20% and 40%. Once the selection of examples was random, we generated 10 different noisy versions of the data sets for each noise level considered.

Half of these data sets, along with their corrupted counterparts are reserved for training the classification techniques, while the remaining data sets are reserved for testing. For dealing with unbalance, both training and test sets were balanced by undersampling the majority class. Moreover, five of such data sets were randomly produced, in order to better evaluate variations due to the data partitions.

Table 1. Summary of data sets characteristics

Data set	#Attributes	#Examples	#Class	#ME	Data set	#Attributes	#Examples	#Class	#ME
abalone	8	4177	28	16.49	newthyroid	5	215	2	83.72
appendicitis	7	106	2	80.18	optdigits	64	5620	10	10.17
balance	4	625	3	46.08	page-blocks	10	5473	5	89.76
banana	2	5300	2	55.16	phoneme	5	5404	2	70.65
car	6	1728	4	70.02	pima	8	768	2	65.10
contraceptive	9	1473	3	42.70	ring	20	7400	2	50.48
ecoli	7	336	8	42.55	saheart	9	462	2	65.36
flare	11	1066	6	31.05	satimage	36	6435	6	23.82
german	20	1000	2	70.00	sonar	60	208	2	53.36
glass	9	214	6	35.51	spambase	57	4601	2	60.59
haberman	3	306	2	73.52	spectfheart	44	267	2	79.40
hayes-roth	4	132	3	38.63	tae	5	151	3	34.43
heart-statlog	13	270	2	55.55	texture	40	5500	11	9.09
ionosphere	34	351	2	64.10	tic-tac-toe	9	958	2	65.34
iris	4	150	3	33.33	titanic	3	2201	2	67.69
kr-vs-kp	36	3196	2	52.22	twonorm	20	7400	2	50.04
led7digit	7	500	10	11.40	vehicle	18	846	4	25.76
lymphography	18	148	4	54.72	vowel	13	990	11	9.09
monk1	6	556	2	50.00	wdbc	30	569	2	62.74
monk2	6	601	2	65.72	wine	13	178	3	39.88
movement-libras	90	360	15	6.66	yeast	8	1484	10	31.19

5.2 Performance in Noisy Data Sets Identification

An interesting result is the verification of the complexity measures behavior for the several noise rates inserted into the data sets. Histograms from the values calculated for each type of noise and some complexity measures were produced. We show in Figure 1 those measures for which we could verify a better ability to distinguish the noise rates: N1 and N3. These measures are based on the use of distances for assessing the complexity of the frontier between the classes and were more affected by the presence of noise.

The classification accuracy values obtained in the identification of the presence of noisy in data sets without noise and with different levels of noise are represented by using a *heatmap* (a graphical representation of the data matrix by a plan of colored rectangles), illustrated in Figure 2. This figure has five groups of columns of blocks, one group for each level of noise. Each column in a group corresponds to a ML technique. Each row in this figure corresponds to one data set. The colour of each box is associated with the accuracy of a particular level of noise in a particular data set.

The box colours in the heatmap range from red (warmer) to blue (cooler) and each colour represents a different accuracy level. The closer the colour is to red, the higher is the accuracy. Thus, the red colour represents the highest accuracy level (between 75% and 100%) and the blue colour represents the lowest accuracy level (between 0% and 25%). According to the heatmap, for a large number

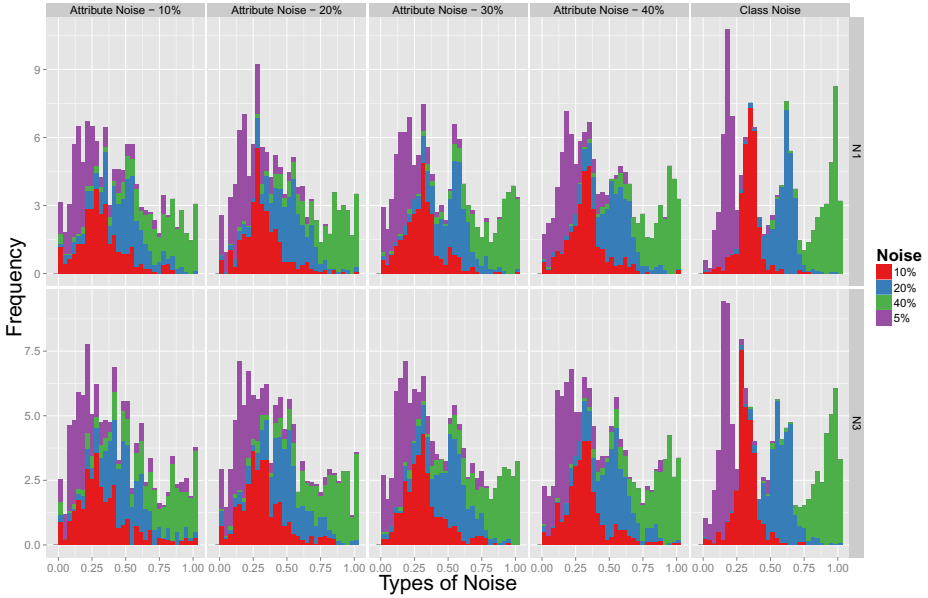


Fig. 1. Histogram of some complexity measure and noise level

data sets and noise levels, the accuracy rates were high. Usually, an increase in the noise level facilitates the identification of its presence. Although in most of the cases a high accuracy was obtained in the identification of noise data sets, there were situations where the accuracy values were low. This occurred for the *appendicitis* and *tic-tac-toe* data sets, where the accuracy values were usually low for attribute and class noise. Nonetheless, for high rates of attribute noise, the accuracy increased for these data sets. Other data sets with some low accuracy values were *saheart*, *specheart*, *haes-roth*, *haberman*, *car* and *balance*.

Table 2 presents the average accuracy values and their standard deviation for each classification algorithm considering all noise identification data sets. It is possible to notice that the higher accuracy values were obtained by the RF classifier. When analyzing the situations with the highest noise level (class noise and predictive attribute noise at a rate of 40%), it is possible to see that, in general, all classifiers presented similar accuracies. For the other cases, larger differences were obtained.

Using the Friedman statistical test with the Nemenyi post-test at 95% of confidence level [26], the following two results were obtained regarding the predictive accuracy of the classifiers investigated: RF results are statistically different from those obtained by 3-NN and SVM; All other pairs of classifiers presented similar predictive accuracy results.

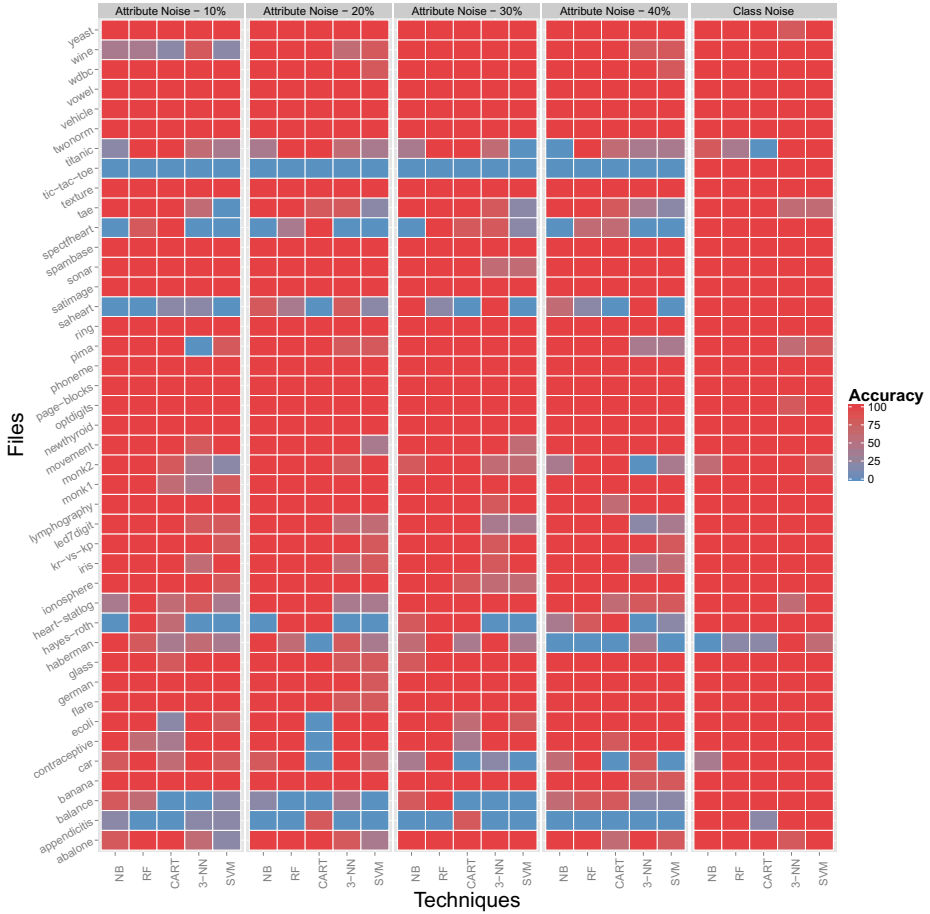


Fig. 2. Accuracy of classifiers for each data set, noise type and level

Table 2. Accuracy of classifiers for the types of noise

Types of Noise	Classifiers				
	NB	RF	CART	3-NN	SVM
Class Noise	95.71 ± 1.06	99.04 ± 1.30	97.14 ± 1.06	95.23 ± 2.38	95.71 ± 3.91
Attr. Noise - 10%	83.33 ± 7.14	91.90 ± 2.12	82.38 ± 6.85	81.90 ± 2.12	83.33 ± 5.32
Attr. Noise - 20%	85.71 ± 3.36	89.04 ± 4.32	86.66 ± 2.71	87.14 ± 3.61	83.33 ± 2.91
Attr. Noise - 30%	87.61 ± 1.06	90.47 ± 3.36	88.09 ± 2.91	83.80 ± 3.91	83.33 ± 5.32
Attr. Noise - 40%	84.28 ± 5.21	94.28 ± 3.61	89.52 ± 6.85	86.19 ± 4.57	86.19 ± 4.25

6 Conclusions

This paper investigated a simple strategy to identify the presence of noise in a data set. A data base composed by several data sets where different noise levels

were artificially injected was created. Each data set was represented by attributes measures extracted to characterize its complexity. The experimental results show that, using this base as input for different learners, it is possible to distinguish noisy from noiseless data sets. The accuracy results regarding this identification were in general high for all types and levels of noise added. However, for some data sets, the classification task was more difficult. This occurred mostly because their original complexity measures values were very similar to those calculated for their noisy counterparts. We intend to investigate further the characteristics of these data sets and the use new measures to better characterize their corruption.

We also plan to employ feature selection strategies to improve the quality of the set of data complexity attributes and eliminating possible irrelevant attributes from the analysis. A higher number of data sets will also be used in the next experiments, in order to improve the results achieved by incorporating more diversity into the base. We shall also investigate new complexity measures for multiclass problems, since some of the used measures are restricted to binary classification problems. Finally, we plan to evaluate a similar method to predict the possible noise level in a data set, which could indicate level of cleaning required.

Acknowledgments. The authors would like to thank FAPESP, CNPq and CAPES for their financial support.

References

1. Wang, R.Y., Storey, V.C., Firth, C.P.: A framework for analysis of data quality research. *IEEE Trans. Knowl. Data Eng.* 7(4), 623–640 (1995)
2. Wu, X.: *Knowledge Acquisition from Databases*. Ablex Publishing Corp. (1995)
3. Maletic, J.I., Marcus, A.: Data cleansing: Beyond integrity analysis. In: *Proc. Conf. Information Quality*, pp. 200–209 (2000)
4. Quinlan, J.R.: Induction of decision trees. *Machine Learning* 1(1), 81–106 (1986)
5. Quinlan, J.R.: The effect of noise on concept learning. In: Michalski, R.S.I., Carbonell, J.G., Mitchell (eds.) *Machine Learning*. Morgan Kaufmann Publishers Inc. (1986)
6. Lorena, A.C., Carvalho, A.C.P.L.F.: Evaluation of noise reduction techniques in the splice junction recognition problem. *Genetics and Molecular Biology* 27(4), 665–672 (2004)
7. Strong, D.M., Lee, Y.W., Wang, R.Y.: Data quality in context. *Commun. ACM* 40(5), 103–110 (1997)
8. Gamberger, D., Lavrac, N., Dzeroski, S.: Noise detection and elimination in data preprocessing: Experiments in medical domains. *Applied Artificial Intelligence* 14(2), 205–223 (2000)
9. John, G.H.: Robust decision trees: Removing outliers from databases. In: *KDD*, pp. 174–179 (1995)
10. Zhao, Q., Nishida, T.: Using qualitative hypotheses to identify inaccurate data. *J. Artif. Intell. Res. (JAIR)* 3, 119–145 (1995)
11. Brodley, C.E., Friedl, M.A.: Identifying and eliminating mislabeled training instances. In: *AAAI/IAAI*, vol. 1, pp. 799–805 (1996)
12. Teng, C.M.: Correcting noisy data. In: *ICML*, pp. 239–248 (1999)

13. Zhu, X., Wu, X., Chen, Q.: Eliminating class noise in large datasets. In: ICML, pp. 920–927 (2003)
14. Zhu, X., Wu, X., Yang, Y.: Error detection and impact-sensitive instance ranking in noisy datasets. In: AAAI, pp. 378–384 (2004)
15. Ho, T.K., Basu, M.: Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.* 24(3), 289–300 (2002)
16. Sáez, J.A., Luengo, J., Herrera, F.: Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. *Pattern Recognition* 46(1), 355–364 (2013)
17. Sluban, B., Gamberger, D., Lavrac, N.: Ensemble-based noise detection: noise ranking and visual performance evaluation. *Data Mining and Knowledge Discovery* (2013)
18. Zhu, X., Wu, X.: Class noise vs. attribute noise: A quantitative study. *Artif. Intell. Rev.* 22(3), 177–210 (2004)
19. Orriols-Puig, A., Maciá, N., Ho, T.K.: Documentation for the data complexity library in C++. Technical report, La Salle - Universitat Ramon Llull (2010)
20. Heckerman, D.: A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research (1995)
21. Breiman, L.: Random forests. *Machine Learning* 45(1), 5–32 (2001)
22. Breiman, L., Freidman, J.H., Olshen, R.A., Stone, C.J.: *Classification and Regression Trees*. Wadsworth (1984)
23. Mitchell, T.M.: *Machine Learning*, 1st edn. McGraw Hill series in computer science. McGraw-Hill (1997)
24. Vapnik, V.N.: *The nature of Statistical learning theory*. Springer (1995)
25. Bache, K., Lichman, M.: *UCI machine learning repository* (2013)
26. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* 7, 1–30 (2006)

Density-Based Clustering in Cloud-Oriented Collaborative Multi-Agent Systems

Jelena Fiosina* and Maksims Fiosins

Clausthal University of Technology,
Institute of Informatics,
Julius-Albert Str. 4, D-38678, Clausthal-Zellerfeld, Germany
{Jelena.Fiosina,Maksims.Fiosins}@gmail.com

Abstract. The development of new reliable data processing and mining methods based on the synergy between cloud computing and the multi-agent paradigm is of great importance for contemporary and future software systems. Cloud computing provides huge volumes of data and computational resources, whereas the agents make the system components more autonomous, cooperative, and intelligent. This creates the need and gives a very good basis for the development of data analysis, processing, and mining methods to enhance the new agent-based cloud computing (ABCC) architecture. Ad-hoc networks of virtual agents are created in the ABCC architecture to support the dynamic functionality of provided services, and data processing methods are very important at the input data processing and network parameter estimation stage. In this study, we present a decentralized kernel-density-based clustering algorithm that fits with the general architecture of ABCC systems. We conduct several experiments to demonstrate the capabilities of the new approach and analyse its efficiency.

Keywords: Cloud computing architecture, distributed data processing and mining, multiagent systems, decentralized clustering, kernel density estimation.

1 Introduction

Cloud computing (CC) is developing rapidly due to new communication and mobile technologies, and it has been introduced recently as a new model for delivering computational resources over a network. Motivated by future Internet technologies such as the Internet of Things, it provides end users with simple on-demand access to services, such as applications or databases, through lightweight mobile applications. Simultaneously, the complexity of the infrastructure is hidden in the cloud, which allows users to focus on their goals instead of the infrastructure complexity [11].

* The research leading to these results has received funding from the EU 7th Framework Programme (FP7/2007-2013) under grant agreement No. PIEF-GA-2010-274881.

It should be noted that cloud-based systems are complex systems, distributed by regions, services, and providers. A popular paradigm for modelling complex distributed systems is the multi-agent system (MAS). Agents in an MAS are autonomous and intelligent, and capable of cooperating with each other and interacting with the environment.

The synergy between MAS and CC models (Fig. 1) reveals new perspectives for developing future intelligent information and management systems. CC provides elastic services, high performance, and scalable data storage to a large and increasing number of users [9]. MAS provides intelligent system behaviour and adaptive mechanisms for data processing, decision-making, and learning to better satisfy user needs as well as intelligent interaction and cooperation mechanisms for dealing with system distribution. In other words, MAS makes CC more intelligent, and CC makes MAS more powerful and accessible.

Agent-based cloud computing (ABCC) [4],[11] is a new research direction that enhances existing complex systems modelled using MASs with new modern technologies from communication and data analysis fields to make corresponding applications more intelligent. Talia [11] considered the implementation of CC with software agents to create intelligent cloud services. CC can offer a very powerful, reliable, predictable, and scalable computing infrastructure for the execution of an MAS implementing complex agent-based applications for modelling and simulation. On the other hand, software agents can be used as the basic components for implementing intelligence in clouds, making them more adaptive, flexible, and autonomic in resource management, service provisioning, and running large-scale applications.

The high availability of mobile devices with sensors and permanent Internet connections means that huge amounts of data are available on CC systems. The appropriate use of such data can create a complete picture of the environment for agents in an MAS, enabling optimal decisions. Hence, the novel mechanisms and algorithms for data processing and mining in ABCC are of high importance [8], [5]. Large amounts of data must be found, collected, aggregated, processed, and analysed for optimal decision-making and behaviour strategy determination. Although information is virtually centralized by cloud technologies, it should be managed in a decentralized fashion, creating challenges for research in this area.

In our previous work, we considered decentralized data processing models, such as regression forecasting and change-point analysis, and applied them for optimal decision making in an MAS, such as optimal route selection [3] or lane/speed adaption [5] in traffic. We demonstrated that appropriate data coordination mechanisms can provide almost the same forecasting accuracy as a model with central authority [2].

In this study, we focus on decentralized data clustering, which is an important data pre-processing step in cloud data repositories. By grouping similar data together, it is possible to construct more precise forecasting models as well as use only typical data representatives in the decision-making process. Complex forms

of clusters require non-parametric, computationally intensive approaches such as kernel-density (KD) [6] clustering (Fig. 1). Fast KD clustering was described by Hinnenburg and Gabriel [7]. The distributed (with central authority) version of KD clustering (KDEC scheme) was considered in [8]. Another graph-oriented decentralized clustering method not based on KD was presented in [10].

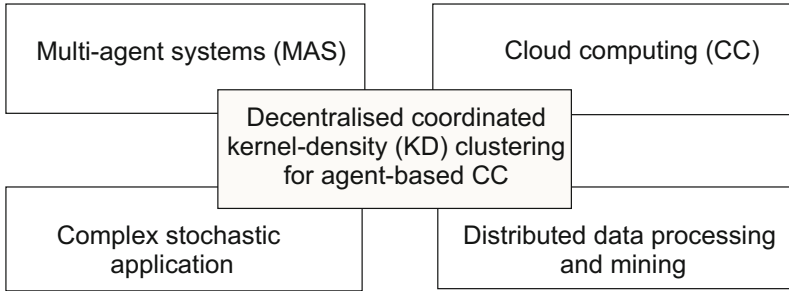


Fig. 1. Synergy of cloud computing and multiagent systems to meet decentralized density-based clustering for some application

The decentralised KD clustering algorithm was motivated by and developed for use in ABCC. The developed algorithm is an extension of the approach [7] for the multivariate case and developing a data coordination scheme based on the transmission of the number of nearest data points from the same cluster.

The remainder of this paper is organized as follows. Section 2 introduces KD clustering. In Section 3, we develop the decentralised cooperative KD clustering algorithm. In Section 4, we conduct several experiments and analyse the efficiency of the suggested approach. The last section presents the conclusion and discusses the opportunities for future work.

2 Kernel Density (KD) Clustering

Now let us formulate the clustering problem and describe the KD clustering algorithm. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, $\mathbf{x}_i \in R^d$ be a dataset to be clustered into k non-overlapping subsets S_1, S_2, \dots, S_k .

Non-parametric clustering methods are well suited for exploring clusters without building a generative model of the data. KD clustering consists of a two-step procedure: estimation and optimisation. During the estimation step, the probability density of the data space is directly estimated from data instances. During the optimisation step, a search is performed for densely populated regions in the estimated probability density function.

Let us formalize the estimation step. The density function is estimated by defining the density at any data object as being proportional to a weighted sum

of all objects in the data-set, where the weights are defined by an appropriately chosen kernel function [8].

A KD estimator is

$$\hat{\psi}^{[X]}(\mathbf{x}) = \frac{1}{N} \sum_{\mathbf{x}_i \in \mathbf{X}} |\mathbf{H}|^{-1} K(\mathbf{H}^{-1} \|\mathbf{x} - \mathbf{x}_i\|) = \frac{1}{N} \sum_{\mathbf{x}_i \in \mathbf{X}} K_{\mathbf{H}}(\|\mathbf{x} - \mathbf{x}_i\|), \quad (1)$$

where $\|\mathbf{x} - \mathbf{x}_i\|$ is a distance between \mathbf{x}_i and \mathbf{x} , \mathbf{H} is a *bandwidth* matrix, $K(\mathbf{x})$ is a kernel function, $K_{\mathbf{H}}(\bullet) = |\mathbf{H}|^{-1} K(\mathbf{H}^{-1}\bullet)$ [6].

$K(\mathbf{x})$ is a real-valued, non-negative function on R^d and has finite integral over R^d . We use the multivariate Gaussian function in our study: $K(\mathbf{x}) = (2\pi)^{-d/2} \exp(-\frac{1}{2}\mathbf{x}^T \mathbf{x})$. The bandwidth matrix \mathbf{H} is a $d \times d$ positive-definite matrix that controls the influence of data objects and smoothness of the estimate. If no information is available with regard to correlation between factors, a diagonal matrix $\mathbf{H} = \text{diag}(h_1, \dots, h_d)$ can be used.

Let us now formalize the optimisation step. This step detects maxima of KD and groups all of the data objects in their neighbourhood into corresponding clusters. We use a hill climbing method for KD maxima estimation with Gaussian kernels (DENCLUE2) [7] and modify the technique for the multivariate case. This method converges towards a local maximum and adjusts the step size automatically at no additional costs. Other optimization methods (DENCLUE) [7] require more steps and additional computations for step size detection.

Each KD maximum can be considered as the centre of a point cluster. With centre-defined clusters, every local maximum of $\hat{\psi}(\cdot)$ corresponds to a cluster that includes all data objects that can be connected to the maximum by a continuous, uphill path in the function of $\hat{\psi}(\cdot)$. Such centre-defined clusters allows for arbitrary-shaped clusters to be detected, including non-linear clusters. An arbitrary-shape cluster is the union of centre-defined clusters that have maxima that can be connected by a continuous, uphill path.

The goal of the hill climbing procedure is to maximize the KD $\hat{\psi}^{[X]}(\mathbf{x})$. By setting the gradient $\nabla \hat{\psi}^{[X]}(\mathbf{x})$ of KD to zero and solving the equation $\nabla \hat{\psi}^{[X]}(\mathbf{x}) = 0$ for \mathbf{x} , we get:

$$\mathbf{x}^{(l+1)} = \frac{\sum_{\mathbf{x}_i \in \mathbf{X}} K_{\mathbf{H}}(\|\mathbf{x}^{(l)} - \mathbf{x}_i\|) \mathbf{x}_i}{\sum_{\mathbf{x}_i \in \mathbf{X}} K_{\mathbf{H}}(\|\mathbf{x}^{(l)} - \mathbf{x}_i\|)}. \quad (2)$$

The formula (2) can be interpreted as a normalized and weighted average of the data points. The weights for each data point depend on the influence of the corresponding kernels on $\mathbf{x}^{(l)}$. Hill climbing is initiated at each data point $\mathbf{x}_i \in \mathbf{X}$ and is iterated until the density does not change, i.e. $|\hat{\psi}^{[X]}(\mathbf{x}_i^{(l)}) - \hat{\psi}^{[X]}(\mathbf{x}_i^{(l-1)})|/\hat{\psi}^{[X]}(\mathbf{x}_i^{(l)}) \leq \epsilon$, where ϵ is a small constant. The end point of the hill climbing algorithm is denoted by $\mathbf{x}_i^* = \mathbf{x}_i^{(l)}$, corresponding to a local maximum of KD.

Now we should determine a cluster for \mathbf{x}_i . Let $\mathbf{X}^c = \{\mathbf{x}_1^c, \mathbf{x}_2^c, \dots\}$ be an ordered set of already identified cluster centres (initially, we suppose $\mathbf{X}^c = \emptyset$). First we find an index of the nearest cluster centre from \mathbf{x}_i^* in the set \mathbf{X}^c :

$$nc(\mathbf{x}_i^*) = \arg \min_{j: \mathbf{x}_j^c \in \mathbf{X}^c} \|\mathbf{x}_j^c - \mathbf{x}_i^*\|.$$

If the nearest cluster centre is close to \mathbf{x}_i^* , then the point \mathbf{x}_i is included in this cluster; otherwise, the point is used as a cluster centre to form a new cluster

$$\Lambda(\mathbf{x}_i) \leftarrow \begin{cases} nc(\mathbf{x}_i^*) & \text{if } \frac{\|\mathbf{x}_{nc(\mathbf{x}_i^*)}^c - \mathbf{x}_i^*\|}{\|\mathbf{x}_i^*\|} \leq \delta, \\ |\mathbf{X}^c| + 1 & \text{otherwise.} \end{cases}$$

where δ is a small constant and $\Lambda(x)$ is a class labeling function. In the second case, we also create a new cluster centre: $\mathbf{X}^c \leftarrow \mathbf{X}^c \cup \{\mathbf{x}_i^*\}$.

3 Decentralized KD Clustering

In this section, we describe the cooperation for sharing the clustering experience among the agents in a network. While working with streaming data, one should take into account two main facts. The nodes should coordinate their clustering experience over some previous sampling period and adapt quickly to the changes in the streaming data, without waiting for the next coordination action.

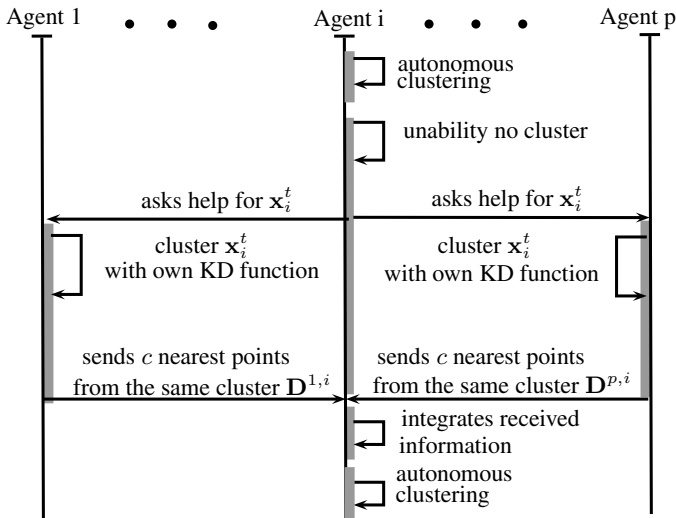


Fig. 2. Interaction between agents

Let us first discuss the cooperation technique (Fig. 2). We introduce the following definitions. Let $\mathbf{A} = \{A^j | 1 \leq j \leq p\}$ be a group of p agents. Each A^j has a local dataset $\mathbf{D}^j = \{\mathbf{x}_t^j | t = 1 \dots, N^j\}$, where $\mathbf{x}_t^j \in R^d$. In order to underline the dependence of the KD function (1) on the local dataset of A^j , we denote the KD function by $\hat{\psi}^{[\mathbf{D}^j]}(\mathbf{x})$.

Consider a case when some agent A^i is unable to classify (after optimisation has formed a new or small cluster) some future data point \mathbf{x}_t^i because it does not have sufficient data in the neighbourhood of this point. It sends the data point \mathbf{x}_t^i to the other neighbouring agents. Each A^j that has received the request classifies \mathbf{x}_t^i using its own KD function $\hat{\psi}^{[\mathbf{D}^j]}(\mathbf{x}_t^i)$ and performs the optimisation step to identify the cluster for this point. Let $n_{j,i}$ be a number of points in the cluster of \mathbf{x}_t^i , not including \mathbf{x}_t^i itself. In the case of successful clustering ($n_{j,i} > 0$), A^j forms an answer $\mathbf{D}^{j,i}$ with c nearest points to the requested data point from the same cluster as \mathbf{x}_t^i (or all points from the cluster, if $n_{j,i} \leq c$). Let $c_{j,i}$ be a number of points in the answer $\mathbf{D}^{j,i}$. The agent A^j sends $\mathbf{D}^{j,i}$ together with $c_{j,i}$ and $n_{j,i}$ to A^i .

After receiving all the answers, A^i forms a new dataset $\hat{\mathbf{D}}^{j,i}$. The next problem is the updating of the KD function of A^i with respect to the new knowledge $\mathbf{D}^{j,i}$. Density estimates (1) of each agent are additive, i.e. the aggregated density estimate $\hat{\psi}^{[\hat{\mathbf{D}}^i]}(\mathbf{x})$ can be decomposed into the sum of the local density estimates, one estimate for every dataset $\mathbf{D}^{j,i}$:

$$\hat{\psi}^{[\hat{\mathbf{D}}^i]}(\mathbf{x}) = w_i \cdot \hat{\psi}^{[\mathbf{D}^i]}(\mathbf{x}) + \frac{(1 - w_i)}{\sum_{A^j \in \mathbf{G}^i} n_{j,i}} \sum_{A^j \in \mathbf{G}^i} n_{j,i} \hat{\psi}^{[\mathbf{D}^{j,i}]}(\mathbf{x}), \tag{3}$$

where w_i is a weight used for the agent's own local observations.

After updating its KD function, A^i can perform a hill-climbing optimisation procedure to identify clusters in its local data space.

To measure the **clustering similarity** [1] among the agents $A^i \in \mathbf{A}$ we use the following representation of a class labeling by a matrix C with components:

$$C_{i,j} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ belong to the same cluster and } i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

Let two labelings have matrix representations $C^{(1)}$ and $C^{(2)}$, respectively. We define a dot product that computes the number of pairs clustered together $\langle C^{(1)}, C^{(2)} \rangle = \sum_i \sum_j C_{i,j}^{(1)} C_{i,j}^{(2)}$. The Jaccard's similarity measure can be expressed as

$$J(C^{(1)}, C^{(2)}) = \frac{\langle C^{(1)}, C^{(2)} \rangle}{\langle C^{(1)}, C^{(1)} \rangle + \langle C^{(2)}, C^{(2)} \rangle - \langle C^{(1)}, C^{(2)} \rangle}. \tag{4}$$

4 Experimental Results

We consider a clustering model with decentralised coordinated architecture. The agents made a local clustering of their observations and used cooperative

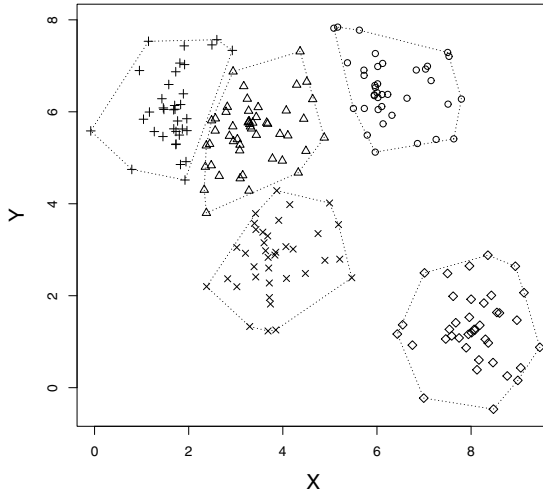


Fig. 3. All observations form clusters

mechanisms to adjust the cluster information according to those of other agents. The amount of information transmitted was lower than that in the centralised model, because it requires no transmission of all global data.

We simulate 10 agents with the initial experience, which varies in the range from 10 to 100 observations. Most simulation experiments ran for 200 time units. For our experiments we assume that all observations are homogeneous and the agents try to estimate the same clusters. The initial global two-dimensional sample data are presented in Fig. 3, where one can see five clusters. The points are located at the normally distributed distances from the cluster centres. Agents take random subsets from this global dataset and try to estimate the clusters by only part of observations. One data synchronization step is demonstrated in Fig. 4. The agent that has difficulties with a point sends a help request. The helping agent clusters the point using its own data and detects corresponding cluster. It sends an answer from three nearest points in the cluster back to the requesting agent. The requesting agent adds received data to its own and makes new clustering. This allows to improve clustering similarity of these two agents from 0.011 to 0.037 as well as clustering similarity of the requesting agent with the 'ideal' clustering from 0.004 to 0.006.

We demonstrate now a system dynamics for a different number of transmitted points (Fig. 5). Clustering similarity (right) increases faster for a bigger number of the transmitted points, but the number of communication events (left) decreases faster. However, we note that one communication event is more expensive for a bigger number of transmitted points, but supplies more information.

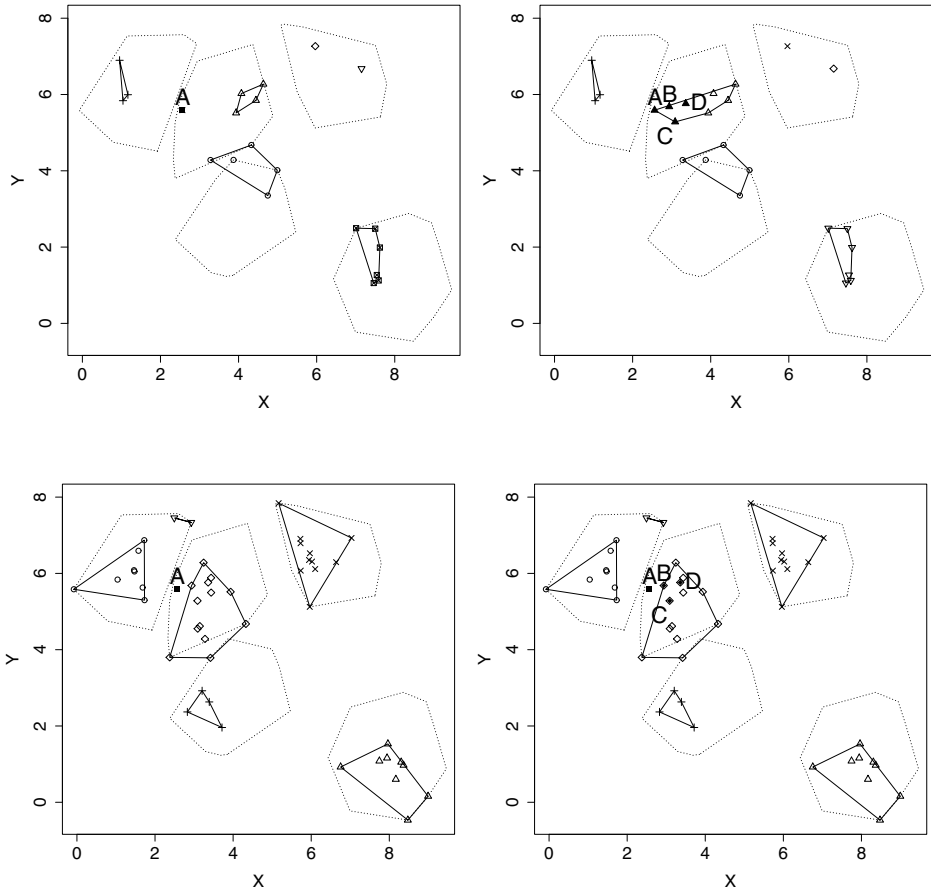


Fig. 4. A communication step between the requesting (top) and helping (bottom) agents. The requesting agent asks for help for point A (top left), the helping agent finds a corresponding cluster (bottom left), and sends the nearest three points B, C, D to the helping agent (bottom right). The helping agent adds the points to its data and makes new clustering (top right).

Quality of the agent models was also checked by a cross-validation technique (Fig. 6) at the beginning (left) and at the end (right) of the simulation. These histograms show a probability distribution of a similarity at the beginning and at the end of the simulation process. One can see that the similarity peak moves to bigger value during the coordination procedure.

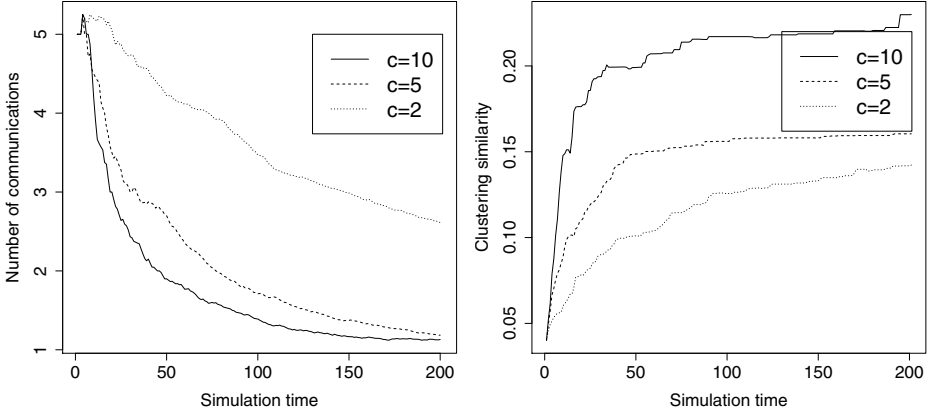


Fig. 5. A number of communication events (left) and similarity of agents’ clusters (right) over time

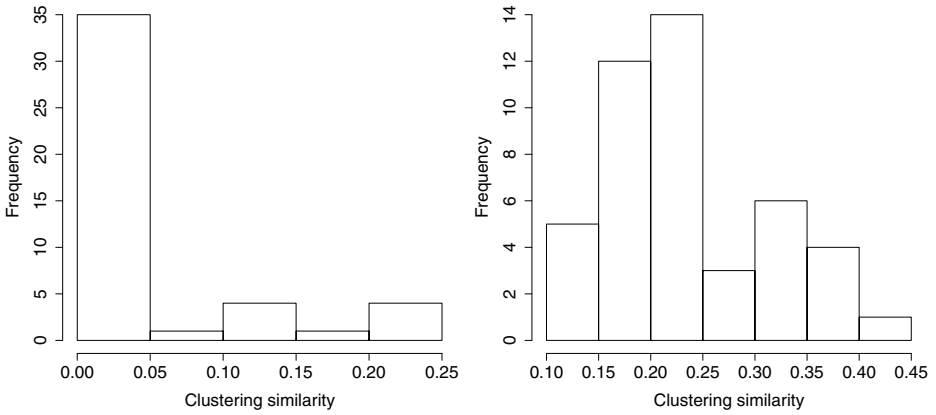


Fig. 6. Frequencies at the beginning (left) and at the end (right) of simulation

5 Future Work and Conclusions

We developed the coordinated decentralized kernel-density clustering approach for agent-based cloud computing architecture. The data coordination scheme is based on the transmission of a several nearest points from the same cluster. An experimental validation of the developed algorithm was also performed. Demonstrated algorithms of collaborative clustering can be applied in cloud-based systems from various domains (e.g. traffic, logistics, energy). Our future work is devoted to the development of new coordination schemes in proposed decentralised clustering approach as well as the application of this algorithm to real-world data.

References

1. Ben-Hur, A., Elisseeff, A., Guyon, I.: A stability based method for discovering structure in clustered data. In: Pacific Sym. on Biocomputing 7, pp. 6–17 (2002)
2. Fiosina, J., Fiosins, M.: Cooperative regression-based forecasting in distributed traffic networks. In: Memon, Q.A. (ed.) Distributed Network Intelligence, Security and Applications, ch. 1, pp. 3–37. CRC Press, Taylor and Francis Group (2013)
3. Fiosina, J., Fiosins, M.: Selecting the shortest itinerary in a cloud-based distributed mobility network. In: Omatu, S., Neves, J., Rodriguez, J.M.C., Paz Santana, J.F., Gonzalez, S.R. (eds.) Distrib. Computing & Artificial Intelligence. AISC, vol. 217, pp. 103–110. Springer, Heidelberg (2013)
4. Fiosina, J., Fiosins, M., Müller, J.P.: Mining the traffic cloud: Data analysis and optimization strategies for cloud-based cooperative mobility management. In: Casillas, J., Martínez-López, F.J., Vicari, R., De la Prieta, F. (eds.) Management Intelligent Systems. AISC, vol. 220, pp. 25–32. Springer, Heidelberg (2013)
5. Fiosins, M., Fiosina, J., Müller, J., Görmer, J.: Agent-based integrated decision making for autonomous vehicles in urban traffic. In: Demazeau, Y., Pěchouček, M., Corchado, J.M., Pérez, J.B. (eds.) Adv. on Prac. Appl. of Agents and Mult. Sys. AISC, vol. 88, pp. 173–178. Springer, Heidelberg (2011)
6. Härdle, W., Müller, M., Sperlich, S., Werwatz, A.: Nonparametric and Semiparametric Models. Springer, Heidelberg (2004)
7. Hinneburg, A., Gabriel, H.-H.: DENCLUE 2.0: Fast clustering based on kernel density estimation. In: Berthold, M., Shawe-Taylor, J., Lavrač, N. (eds.) IDA 2007. LNCS, vol. 4723, pp. 70–80. Springer, Heidelberg (2007)
8. Klusch, M., Lodi, S., Moro, G.: Agent-based distributed data mining: The KDEC scheme. In: Klusch, M., Bergamaschi, S., Edwards, P., Petta, P. (eds.) Intelligent Information Agents. LNCS (LNAI), vol. 2586, pp. 104–122. Springer, Heidelberg (2003)
9. Armbrust, M., et al.: A view of cloud computing. Communications of the ACM 53(4), 50–58 (2010)
10. Ogston, E., Overeinder, B., van Steen, M., Brazier, F.: A method for decentralized clustering in large multi-agent systems. In: Proc. of 2nd Int. Conf. on Autonomous Agents and Multiagent Systems, pp. 789–796 (2003)
11. Talia, D.: Cloud computing and software agents: Towards cloud intelligent services. In: Proc. of the 12th Workshop on Objects and Agents, vol. 741, pp. 2–6 (2011)

A Hybrid Genetic Algorithm with Variable Neighborhood Search Approach to the Number Partitioning Problem

Levente Fuksz¹ and Petrica C. Pop²

¹ Indeco Soft, Baia Mare, Romania

levi.fuksz@yahoo.com

² Dept. of Mathematics and Computer Science, North Univ. Center of Baia Mare, Technical University of Cluj-Napoca, Str. Victoriei, 430122, Baia Mare, Romania

petrica.pop@ubm.ro

Abstract. The article presents a novel approach for solving the number partitioning problem. Our approach combines the use of genetic algorithms (GA) and Variable Neighborhood Search (VNS) resulting a new highly scalable hybrid GA-VNS (Genetic Algorithm with Variable Neighborhood Search), which runs the GA as the main algorithm and the VNS procedure for improving individuals within the population. The preliminary experimental results indicate that the GA-VNS hybrid algorithm performs significantly better, in terms of the solution quality, in comparison to the existing heuristic algorithms and to the pure GA for solving the number partitioning problem.

Keywords: number partitioning problem, genetic algorithms, Variable Neighborhood Search, hybrid algorithms.

1 Introduction

The number partitioning problem is a classical, challenging and surprisingly difficult problem in combinatorial optimization. Given a set S of n integers, the two-way number partitioning problem, denoted by TWNPP, asks for a division of S into two subsets such that the sums of numbers in each subset should be equal or are close to be equal.

Though the number partitioning problem is NP-complete (see [5]), there have been proposed heuristic algorithms that solve the problem in many instances either optimally or approximately. This is one of the reasons for which Hayes has called the number partitioning problem "The Easiest Hard Problem" [8].

A variation of the number partitioning problem is the set partitioning problem that requires partitioning a set of n numbers into p subsets such that the difference between the maximum and the minimum subset sums is minimized. Additional variants are considering cardinality constraints, namely the cardinality of the subsets should be balanced, for more information on these variants we refer to [14,17].

The number partitioning problem captioned a lot of attention due to its theoretical aspects and properties and important real-world applications in multi-processor scheduling, the minimization of VLSI circuit size and delay, public key cryptography, etc. For a more detailed description of the applications we refer to [3].

There are several ways to solve the TWNPP in exponential time in n . The most naive algorithm would be to cycle through all the subsets of n numbers and for every possible subset S_1 and for its corresponding complementary $S_2 = S \setminus S_1$ calculate their sums. Obviously, this algorithm is impracticable for large instances, its time complexity being $O(2^n)$. A better exponential time algorithm which runs in $O(2^{n/2})$ was described by Horowitz and Sahni [9].

Various heuristic algorithms have been developed for solving the TWNPP including: a natural greedy algorithm obtained by sorting the numbers in decreasing order and then assigning each number in turn to the subset with the smaller sum so far; a complete greedy algorithm described by Korf [12] where based on a binary tree each level assigns a different number and each branch point alternately assigns that number to one subset or the other; the set differencing heuristic introduced by Karmarkar and Karp [11] that repeatedly replaces the two largest numbers with their difference, inserting the new number in the sorted order until there is only one number left which is the final partition difference, the complete Karmarkar-Karp algorithm developed by Korf [12], a hybrid recursive algorithm obtained by combining several existing algorithms with some new extensions developed by Korf [13], etc.

Several metaheuristic approaches have been proposed for solving the two-way number partitioning problem including a Simulated Annealing algorithm by Johnson et al. [10], genetic algorithm by Ruml et al. [16], GRASP by Arguello et al. [1], Tabu Search by Glover and Laguna [6], memetic algorithm by Berretta et al. [2], etc.

The aim of this paper is to describe a novel use of genetic algorithms with the goal of solving the two-way number partitioning problem and as well a new hybrid approach to the problem, that combines the use of genetic algorithms (GA) and Variable Neighborhood Search (VNS). The resulting hybrid approach GA-VNS runs the GA as the main algorithm and uses the VNS procedure in order to improve the individuals in the population. The results of preliminary computational experiments are presented, analyzed and compared with the previous heuristic methods. The results reveal that our proposed hybrid GA-VNS algorithm performs significantly better in comparison to the existing approaches and the pure GA.

2 Definition of the TWNPP and Its Extensions

Given a set of n positive integer numbers

$$S = \{a_i \mid i \in \{1, \dots, n\}\}$$

then the two-way number partitioning problem consists in splitting the elements of S into two sets, S_1 and S_2 such that

1. $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \emptyset$;
2. the sums of elements in the subsets S_1 and S_2 are equal or almost equal.

Equivalently, the TWNPP asks for finding a subset $A \subset \{1, \dots, n\}$ such that the discrepancy:

$$D(A) = \left| \sum_{i \in A} a_i - \sum_{i \notin A} a_i \right|$$

is minimized.

A partition is called perfect if the minimum discrepancy is 0 when the sum of all n integers in the original set is even, or 1 when the sum is odd.

The TWNPP can be generalized to the case where a set of positive integer numbers is partitioned into a given number of subsets rather than into two subsets. We will call this generalization the multi-way number partitioning problem and can be defined as follows.

Let again S be a set of n positive integer numbers and $p \in \mathbb{N}$, $p \geq 2$, then the multi-way number partitioning problem consists in splitting the elements of S into p subsets, S_1, S_2, \dots, S_p such that

1. $S_1 \cup S_2 \cup \dots \cup S_p = S$ and $S_i \cap S_j = \emptyset$, for all $i, j \in \{1, \dots, p\}$ and $i \neq j$;
2. the sums of elements in the subsets S_1, S_2, \dots, S_p are equal or almost equal.

In particular, if the the set of positive integer numbers is partitioned into two subsets we get the TWNPP. For partitioning into more than two subsets, the objective function to be minimized is the greatest difference between maximum and minimum subset sums.

3 Hybrid Approach Outline

In this section we outline two genetic algorithms for solving the number partitioning problem: the first one is a "pure" genetic algorithm and forms as well the basis of the second one: a hybrid algorithm obtained from the combination of the GA with the local search (LS) ability of Variable Neighborhood Search (VNS).

3.1 The Genetic Algorithm

The Genetic Algorithms (GA) were introduced by Holland in the early 1970s, and were inspired by Darwin's theory. The idea behind GA is to model the natural evolution by using genetic inheritance together with Darwin's theory. GA have seen a widespread use among modern metaheuristics, and several applications to combinatorial optimization problems have been reported.

Next we give the description of our genetic algorithm for solving the two-way number partitioning problem.

Representation

In order to represent a potential solution to the TWNPP, we used a binary representation where every chromosome is a fixed size (n -dimensional vector) ordered string of bits 0 or 1, identifying the set of partition as assigned to the numbers. This representation ensures that the set of vectors belonging to the set S is partitioned into two subsets S_1 and S_2 .

Initial Population

The construction of the initial population is of great importance to the performance of GA, since it contains part of the building blocks the final solution is made of, which is then combined by the crossover operator. If at the beginning the GAs used mainly randomly generated initial population there is an increasing interest to seed the initial population with some good candidate solutions or partial solutions in order to provide some hints concerning the evolution process.

Population seeding is an interesting and important aspect of GA, because the final solution quality and the convergence of the algorithm may be improved by inserting non-random chromosomes (seeds), rather fitter than the average random chromosomes, into the initial population. For more information concerning population seeding we refer to [4].

We have been carried out experiments with two different ways of generating the initial population:

- 1) A common method of generating the population is random generation. Each gene for a chromosome assumes a value of 1 with probability p and a value of 0 with probability $1 - p$, quite commonly $Pr(X = 1) = 0.5$. This approach is efficient and provides a population covering the feasible region but it may lead to large values of the objective function yielding poor performance of the GA algorithm.
- 2) We considered as well a novel method for generating the initial population: partially randomly and partially based on the problem structure. In this case, we pick randomly a number $q \in \{2, \dots, n\}$ and then for the numbers belonging to $\{2, \dots, q\}$ the genes are generated randomly and the other numbers are partitioned iteratively such that by adding each number we reduce the discrepancy.

Generating the initial population using as well the information about the problem structure, by considering seeded partial solutions, permitted us to reduce the global fitness of the initial population with about 50% in comparison to the randomly generation of the initial population.

The Fitness Value

GAs require a fitness function which allocates a score to each chromosome in the current population. Thus, it can calculate how well the solutions are coded and how well they solve the problem. In our case, the fitness value of the TWNPP, for a given partition of the numbers into two subsets is given by the

corresponding discrepancy. The aim is to find a partition that minimizes the discrepancy.

Selection

Selection is the process used to select individuals for reproduction to create the next generation. This is driven by a fitness function that makes higher fitness individuals more likely to be selected for creating the next generation. We have implemented three different selection strategies: the fitness proportionate selection, the elitist selection and the tournament selection. Several TWNPP were tested and the results show that the tournament selection strategy outperformed the other considered selection strategies, achieving best solution quality with low computing times.

Genetic Operators

Genetic operators are used in genetic algorithms in order to combine the existing solutions into others (crossover-like operators) and to generate diversity (mutation-like operators). The main difference among them is that the latter operate on one chromosome, that is, they are unary operators, while the former are binary operators.

Crossover Operator

During each successive generation, a proportion of the existing population is selected to produce a new generation. The crossover operator requires some strategy to select two parents from previous generation. In our case we selected the two parents using the binary tournament method, where p solutions, called parents, are picked from the population, their fitness is compared and the best solution is chosen for a reproductive trial. In order to produce a child, two binary tournaments are held, each of which produces one parent.

We have experimented a single point crossover. The crossover point is determined randomly by generating a random number between 1 and $n - 1$. We decided upon crossover rate of 85% based on preliminary experiments with different values.

Mutation Operator

Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm may be able to identify better solutions than was previously possible. Mutation is an important part of the genetic search as helps to prevent the population from stagnating at any local optima and its purpose is to maintain diversity within the population and to inhibit the premature convergence.

We consider a mutation operator that changes the new offspring by flipping bits from 1 to 0 or from 0 to 1. Mutation can occur at each bit position in the string with 10% probability.

An important feature of our GA, that increased its performance, is that every time a new population is produced, we eliminate the duplicate solutions.

In our algorithm the termination strategy is based on a maximum number of generations to be run if the optimal solution of the problem is not found or no improvement of the discrepancy value is not observed within 15 consecutive generations.

As we will see in the next Section, our proposed GA is effective in producing good solutions. However, due to the weakness of GAs to intensify the search in promising areas of the solutions space, we will combine our GA with the local search ability of VNS in order to enhance the exploitation ability of GAs.

3.2 The GA-VNS Hybrid Algorithm

Variable neighborhood search (VNS) is quite a recent metaheuristic for solving combinatorial optimization and global optimization problems introduced by Mladenovic and Hansen [7,15]. Its basic idea is a systematic change of neighborhood both within a descent phase to find a local optimum and in a perturbation phase to get out of the corresponding valley. Applications of VNS for solving optimization problems have been rapidly increased in many fields: network design, location theory, vehicle routing, artificial intelligence, engineering, etc.

We used here the same GA as the one described in the previous section, but the difference is that a Variable Neighborhood Search is used along with the genetic algorithm.

Applying local search to all the individuals of a current population will lead to highly time consuming procedure, therefore we selected a subset of individuals in each generation with a specified probability and then the VNS procedure is applied to each of them separately. In the case better individuals are found they are introduced in the current population.

Our VNS algorithm applies 10 types of neighborhoods, denoted by \mathcal{N}_i , $i \in \{1, \dots, 10\}$. The neighborhoods are implemented as inversions of bits (representing either 0 or 1) within the chromosome with positions generated randomly. The ten neighborhoods correspond to the number of bits which are inverted $i \in \{1, \dots, 10\}$. For each neighborhood the following repetitive loop is applied: we choose randomly the positions within the chromosome made up of bits and then we inverse the corresponding genes by logical negation.

The first neighborhood is the set of candidate solutions that have one bit difference against the current solution. We select randomly an entry from the string representation and then inverse the value of the corresponding gene, meaning that we assign an integer from one set to the other one. If by changing the value of a gene, we obtain a neighbor having a lower cost than the current solution, than the neighbor becomes the new current solution and the search proceeds. The search process continues until no better solution is found in the neighborhood, then the search switches to the second neighborhood, which consists of candidate solutions having exactly two bits difference against the current solution. This new neighborhood is examined in order to find an improvement solution and the search continues till a better solution cannot be found.

Then the search switches to the new neighborhood and the process goes on iteratively.

The switching of neighborhoods prevents the search being struck at a local minimum. When there is no better solution found in a current neighborhood, it can be a local optimum, but by changing the neighborhood, it is highly probable that a better solution can be found and the local optimum is skipped.

The described strategy show how to use VNS in descent in order to escape from a local optimum of and now we are interested in finding promising regions for sub-optimal solutions.

Our implementation of the VNS procedure is described in Algorithm 1.

Algorithm 1. Variable Neighborhood Search Framework

Initialization. Select a set of neighborhoods structures $\mathcal{N}=\{\mathcal{N}_l \mid l = 1, \dots, 10\}$; an initial solution x and a stopping criterion

Repeat the following sequence till the stopping criterion is met:

- (1) Set $l = 1$;
- (2) Repeat the following steps until $l = 10$:

Step 1 (Shaking): Generate $x' \in \mathcal{N}_l$ at random;

Step 2 (Local Search): Apply a local search method starting with x' as initial solution and denote by x'' the obtained local optimum ;

Step 3 (Move or not): If the local optimum x'' is better than the incumbent x , then move there ($x \leftarrow x''$) and continue the search with \mathcal{N}_1 otherwise set $l = l + 1$ (or if $l = 10$ set ($l = 1$));

Go back to Step 1.

According to this basic scheme, we can observe that our VNS is a random descent first improvement heuristic.

The algorithm starts with an initial feasible solution x from the selected individuals from the current population and with with the set of the 10 nested neighborhood structures: $\mathcal{N}_1, \dots, \mathcal{N}_{10}$, having the property that their sizes are increasing.

Then a point x' at random (in order to avoid cycling) is selected within the first neighborhood $\mathcal{N}_1(x)$ of x and a descent from x' is done with the local search routine. This will lead to a new local minimum x'' . At this point, there exists three possibilities:

- 1) $x'' = x$, i.e. we are again at the bottom of the same valley and we continue the search using the next neighborhood $\mathcal{N}_l(x)$ with $l \geq 2$;
- 2) $x'' \neq x$ and $f(x'') \geq f(x)$, i.e. we found a new local optimum but which is worse than the previous incumbent solution. Also in this case, we will continue the search using the next neighborhood $\mathcal{N}_l(x)$ with $l \geq 2$;
- 3) $x'' \neq x$ and $f(x'') < f(x)$, i.e. we found a new local optimum but which is better than the previous incumbent solution. In this case, the search is re-centered around x'' and begins with the first neighborhood.

If the last neighborhood has been reached without finding a better solution than the incumbent, than the search begins again with the first neighborhood $\mathcal{N}_1(x)$ until a stopping criterion is satisfied. In our case, as stopping criterion we have chosen a maximum number of iterations since the last improvement.

4 Computational Results

This section presents the obtained results for solving the number partitioning problem. The experiments were carried out on instances obtained using the randomly number generator Random.org.

The testing machine was an Intel Core 2 Quad Q6600 and 3.50 GB RAM with Windows 7 as operating system. The GA and GA-VNS hybrid algorithm have been developed in Microsoft .NET Framework 4 using C #.

Based on preliminary computational experiments, we set the following genetic parameters: the size of the initial population 50 generated half randomly, tournament selection with groups of 7, one-point crossover, mutation probability 10% and maximum number of generations 100.

In the next table we present the obtained computational results using the proposed GA and GA-VNS hybrid algorithm in comparison to the greedy algorithm and Karmarkar-Karp heuristic algorithm [11]. In our experiments, we performed 10 independent runs for each instance.

Table 1. Computational results

Problem instance	Greedy alg.		KK alg.		Results of GA			Results of GA-VNS		
	Best sol.	time	Best sol.	time	Best sol.	Avg. sol.	Avg. time	Best sol.	Avg. sol.	Avg. time
10 (1-100)	0	0	0	0	0	0	0.24	0	0	0.28
10 (1-1000)	22	0	20	0	2	14	0.43	2	2	0.53
10 (1-10000)	2738	0	356	0	14	14.4	0.43	14	14	0.43
10 (1-100000)	43636	0	22876	0	456	456	0.28	456	456	0.28
10 (1-1000000)	12490	0	5202	0	5202	5202	0.34	1494	3718.8	0.218
100 (1-100)	4	0	0	0	0	0	0.99	0	0	0.106
100 (1-1000)	0	0	0	0	0	0	0.96	0	0	0.106
100 (1-10000)	83	0	1	0	1	1	0.187	1	1	0.299
100 (1-100000)	185	0	1	0.15	1	3.8	2.24	1	1	2.726
100 (1-1000000)	3662	0	0	0	6	18.8	3.99	0	18	6.686
10 (100-10000)	626	0	518	0	484	490.8	0.56	106	141.2	1.485
20 (100-10000)	396	0	44	0	2	4.8	0.127	2	2.4	0.365
10 (100-100000)	21155	0	4255	0	1751	1751	0.28	337	337	0.265
20 (100-100000)	21857	0	181	0	13	27	0.18	3	12.2	1.182
10 (1000-10000)	704	0	146	0	84	84	0.28	84	84	0.31
20 (1000-10000)	126	0	4	0	0	0.8	0.162	0	0.8	0.358
10 (1000-100000)	5886	0	5886	0	1236	1236	0.21	1236	1236	0.24
20 (1000-100000)	5482	0	50	0	28	36	1.57	12	28	0.652
10 ($10^9 - 10^6$)	59085	0	8155	0	8155	8155	0.28	1239	5721	0.99
20 ($10^5 - 10^6$)	40478	0	1064	0	144	247.2	1.26	144	221.2	0.468
50 ($10^5 - 10^6$)	11028	0	34	0	12	85.2	2.527	2	33.2	2.87
10 ($10^6 - 10^7$)	1236667	0	275187	0	367087	367087	0.49	62487	62487	0.271
20 ($10^6 - 10^7$)	270954	0	16680	0	1200	1514	0.358	1188	1188	0.218
50 ($10^6 - 10^7$)	304824	0	254	0	32	653.6	2.527	26	109.2	3.45

The first column in the table gives the dimension of the instance followed by the interval from where we have been selected the numbers, the next four columns provide the results and computational times obtained by using the greedy algorithm and Karmarkar-Karp heuristic algorithms and the last six columns give the best solutions, average solutions and average computational times provided by the GA and the hybrid GA-VNS algorithm.

Analyzing the results presented in table 1, we observe that our proposed hybrid GA-VNS heuristic algorithm performs favorable in terms of the solution

quality in comparison with the GA alone, greedy algorithm and Karmarkar-Karp heuristic: in 11 out of 24 instances we have been able to improve the objective function of the TWNPP and for the other instances we obtained the same solutions as those obtained using the GA alone or Karmarkar-Karp algorithm.

We can observe that for small size instances the quality of the solutions provided by our GA-VNS approach is comparable with the GA alone, greedy algorithm and Karmarkar-Karp heuristic, but as the size of instances is increased, our method provides high quality solutions.

The running time of our hybrid GA-VNS is proportional with the number of generations. From table 1, it should be noted that the greedy algorithm and Karmarkar-Karp heuristic are faster than our approaches.

5 Conclusions

In this paper, we considered the two-way number partitioning problem, where a set of integers has to be partitioned into two subsets such that the sums of numbers in each subset should be equal or are close to be equal.

We developed an efficient hybrid approach to the problem that combines the use of genetic algorithms (GA) and Variable Neighborhood Search (VNS). Some important features of our hybrid algorithm are:

- using a novel method for generating the initial population: partially randomly and partially based on the problem structure.
- elimination of the duplicate solutions from each population;
- using the VNS procedure along the GA in order to intensify the search within promising areas of the solution space.

The preliminary computational results show that our hybrid GA-VNS algorithm compares favorably in terms of the solution quality in comparison to the existing approaches and the genetic algorithm alone.

In the future, we plan to assess the generality and scalability of the proposed hybrid heuristic by testing it on more instances and to apply it also in the case of multi-way number partitioning problem.

Acknowledgment. This work was supported by a grant of the Romanian National Authority for Scientific Research, CNCS - UEFISCDI, project number PN-II-RU-TE-2011-3-0113.

References

1. Arguello, M.F., Feo, T.A., Goldschmidt, O.: Randomized methods for the number partitioning problem. *Computers & Operations Research* 23(2), 103–111 (1996)
2. Berretta, R.E., Moscato, P., Cotta, C.: Enhancing a memetic algorithms' performance using a matching-based recombination algorithm: results on the number partitioning problem. In: Resende, M.G.C., Souza, J. (eds.) *Metaheuristics: Computer Decision-Making*. Kluwer (2004)

3. Coffman, E., Lueker, G.S.: Probabilistic Analysis of Packing and Partitioning Algorithms. John Wiley & Sons, New York (1991)
4. Corne, D., Ross, P.: Peckish Initialization Strategies for Evolutionary Timetabling. In: Burke, E.K., Ross, P. (eds.) PATAT 1995. LNCS, vol. 1153, pp. 227–241. Springer, Heidelberg (1996)
5. Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of NP-Completeness. W.H. Freeman, New York (1997)
6. Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Norwell (1997)
7. Hansen, P., Mladenovic, N.: Variable neighborhood search: Principles and applications. *European Journal of Operational Research* 130(3), 449–467 (2001)
8. Hayes, B.: The easiest hard problem. *American Scientist* 90, 113–117 (2002)
9. Horowitz, E., Sahni, S.: Computing partitions with applications to the Knapsack problem. *Journal of ACM* 21(2), 277–292 (1974)
10. Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C.: Optimization by simulated annealing: An experimental evaluation; Part II: Graph coloring and number partitioning. *Operations Research* 39(3), 378–406 (1991)
11. Karmarkar, N., Karp, R.M.: The differencing method of set partitioning, Technical Report UCB/CSD 82/113, Computer Science Division, University of California, Berkeley (1982)
12. Korf, R.E.: A complete anytime algorithm for number partitioning. *Artificial Intelligence* 106(2), 181–203 (1998)
13. Korf, R.E.: A Hybrid Recursive Multi-Way Number Partitioning Algorithm. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, pp. 591–596 (2011)
14. Michiels, W., Aarts, E., Korst, J., van Leeuwen, J., Spijksma, F.C.R.: Computer-assisted proof of performance ratios for the Differencing Method. *Discrete Optimization* 9, 1–16 (2012)
15. Mladenovic, N., Hansen, P.: Variable neighborhood search. *Computers and Operations Research* 24(11), 1097–1100 (1997)
16. Ruml, W., Ngo, J.T., Marks, J., Shieber, S.M.: Easily searched encodings for number partitioning. *Journal of Optimization Theory and Applications* 89(2), 251–291 (1996)
17. Tasi, L.-H.: The modified differencing method for the set partitioning problem with cardinality constraints. *Discrete Applied Mathematics* 63, 175–180 (1995)

Human Activity Recognition and Feature Selection for Stroke Early Diagnosis

José Ramón Villar*, Silvia González, Javier Sedano,
Camelia Chira, and José M. Trejo

¹ University of Oviedo, Gijón, Spain
villarjose@uniovi.es

² Instituto Tecnológico de Castilla y León, Burgos, Spain
{silvia.gonzalez,javier.sedano,camelia.chira}@itcl.es

³ Neurology Department of the Burgos hospital, Burgos, Spain
jtrejogy@gmail.com

Abstract. Human Activity Recognition (HAR) refers to the techniques for detecting what a subject is currently doing. A wide variety of techniques have been designed and applied in ambient intelligence -related with comfort issues in home automation- and in Ambient Assisted Living (AAL) -related with the health care of elderly people. In this study, we focus on the diagnosing of an illness that requires estimating the activity of the subject. In a previous study, we adapted a well-known HAR technique to use accelerometers in the dominant wrist. This study goes one step further, firstly analyzing the different variables that have been reported in HAR, then evaluating those of higher relevance and finally performing a wrapper feature selection method. The main contribution of this study is the best adaptation of the chosen technique for estimating the current activity of the individual. The obtained results are expected to be included in a specific device for early stroke diagnosing.

Keywords: Ambient Assisted Living, Human Activity Recognition, Genetic Fuzzy Finite State Machine, Feature Selection, Genetic Algorithms.

1 Introduction

Stroke is a cerebrovascular disease defined as a circulatory disorder that causes either a temporary or a permanent disorder of one or more areas of the brain. The most common symptom of stroke is loss of the ability to move voluntarily the limbs, either left, right or both. The hand is usually more severely affected compared to the leg [12,8]. Even with a complex and long rehabilitation process, recovery is incomplete [9]. Cerebral infarction makes up to around 85% of all strokes [2,1]. For the rest 15% of cerebral haemorrhages there is no approved treatment, but for the more prevalent cerebral infarction, there is one that can make a big difference: a thrombolytic drug that disrupts the thrombus occluding a cerebral artery. If successful, cerebral tissue will recover and so will function,

* Corresponding author.

but that depends on how fast the treatment is given. In the first one and a half "golden hour", one out of three patients treated will recover to his/her previous life. Unfortunately, reality is very different from what it could be expected: only a minority (5-15% of people suffering a stroke) arrive early enough to actually receive the treatment. Thus, a device that makes people with stroke arrive earlier can make a big difference in reducing death, disability and health costs in thousands of patients each year.

In this sense, determining the current activity of the subject is by no means a previous step before extracting rules that could eventually give advice or generate alarms for possible stroke attacks. Though walking is completely understood [15], activities like sitting or standing can include also eating, reading, etc. In this study we focus on the stroke risk population, which includes adults above 56 years old. Therefore, only the most remarkable activities are considered, as they represent the main part of the activities carried out in everyday life. Evidently, the older the subject is, the higher the probability the subject reduces his/her activities to $\{walking, standing, resting\}$, even though in each activity a wide variety of postures and gestures can be accomplished as well. As mentioned before, the partial or total paralysis detection would generate the corresponding emergency alarm. This research based such detection in the use of accelerometers placed in two bracelets to be wore in each wrist. Nevertheless, we do need to first estimate the current activity of the subject and then estimate whether the lack of motion is a paralysis or not. In a previous study [10], the Genetic Fuzzy Finite State Machine (GFFSM) method for Human Activity Recognition (HAR) [4] was implemented and adapted to the problem of stroke early diagnosing.

In this study, we analyze the different available transformations in the literature and evaluate which are the best suite candidates for being considered the inputs of the GFFSM. Moreover, with the chosen candidates we perform a wrapper feature selection (FS) based on the Steady State Genetic Algorithm (SSGA) FS method [6] but using the GFFSM instead of the KNN classifier. Finally, we evaluate and compare this method with the results from our first study. The main contribution of this paper includes the optimizing of the HAR method and the best feature subset for this task with the sensors placed on the dominant wrists. The remaining of this study is organized as follows. Next section includes the most updated and complete review of the raw acceleration data transformations in the literature, while Sect. 3 the transformation evaluation and the FS method with the GFFSM as classifier is detailed. Sect. 4 deals with the experimentation and the discussion on the results. The study finishes with the Conclusions.

2 Representation of the Raw Acceleration Data

Using triaxial accelerometers induces that the measurement obtained from the sensors, known as raw data (RD, a_i^x , a_i^y and a_i^z ; $a_{i,j \in \{x,y,z\}}$ for the sake of brevity), should be decomposed in the gravity acceleration (G) -that due to the each gravity, g_i^x , g_i^y and g_i^z or $g_{i,j \in \{x,y,z\}}$ - and the body acceleration (BA) -which

is due to the human movement, b_i^x , b_i^y and b_i^z or $b_{i,j \in \{x,y,z\}}$. The capacity of the BA for discriminating among different human gestures is documented [19]. Nevertheless, the literature includes the use of a wide variety of transformations, the most interesting are related in the following, where w stands for the window size -if needed-, and subindexes $i \in \{1, \dots, N\}$ and $j \in \{x, y, z\}$ stand for the number of the sample and the axis, respectively.

1. The *mean, deviation and higher momentum statistics* values for the RD [14] or for the BA [18,19] and the RD *mean absolute deviation* $MAD_j = \frac{1}{w} \sum_{i=1}^w |a_{i,j} - m_j|$ [7,14], where m_j is the mean value of $a_{i,j}$.
2. The *Root Mean Square* $RMS_j = \sqrt{\frac{1}{w} \sum_{i=1}^w |a_{i,j}^2|}$ [7].
3. The *sum of the absolute values* of the BA [10] ($sBA_i = \sum_{j \in \{x,y,z\}} |b_{i,j}|$) and the *vibration of the sensor* (Δ) [18] ($\Delta_i = \sum_{j \in \{x,y,z\}} a_{i,j}^2 - g_{i,j}^2 \sim \sum_{j \in \{x,y,z\}} b_{i,j}^2$) and the *tilt of the body* ($tilt_i = |a_i^y| + |a_i^z|$) [4]. The two former transformations were designed to detect whether the sensor register no movement at all, as fixed to an steady object, while the latter is assumed if the sensor axes correspond with the body axes.
4. The *Signal Magnitude Area* $SMA = \frac{1}{w} \cdot \sum_{i=1}^w (|b_i^x| + |b_i^y| + |b_i^z|)$ [18,3,19] discriminating between gravity acceleration and BA.
5. The *Amount of Movement* $AM_i = \sum_{v \in \{x,y,z\}} |max_{t=i+1}^{i+w}(b_t^v) - min_{t=i+1}^{i+w}(b_t^v)|$ [4]: calculated as the maximum difference among the values of BA within the sliding window.
6. Delta coefficients for estimating the first order time derivate of each of the G signal components [3]: $\Delta g_t^{\{x,y,z\}} = \sum_{d=-D}^D d \cdot g_{t+d}^{\{x,y,z\}} / \sum_{d=-D}^D d^2$, where the shift D is parameterized to the algorithms and $g_t^{\{x,y,z\}}$ stands for each of the three axis G components.
7. Shifted Delta Coefficients (SDC) for estimating the first order time derivate of each of the BA signal components in the vicinity of the current time stamp [3]: $\Delta b_{t+i \cdot P}^{\{x,y,z\}} = \frac{\sum_{d=-D}^D d \cdot b_{t+i \cdot P+d}^{\{x,y,z\}}}{\sum_{d=-D}^D d^2}$, where $b_t^{\{x,y,z\}}$ stands for each of the three axis BA components, N is the number of base features from which they are calculated, D is the same D as in the delta calculations, P is the distance between samples and K is the number of samples taken.
8. Average Energy (AE) [5,18,19]: calculated as the sum of the squared discrete FFT component magnitudes of the signal in a window of a fixed size. This features allows to discriminate between static and dynamic activities. It is calculated for each axis; the aggregation or the average over the three axes is commonly used [19].
9. The correlation between axes [5]: calculated for each pair of axes as the ratio of the covariance and the product of the standard deviations. This feature is useful to discriminate one dimensional activities if your sensory is placed accordingly. As stated in [19], this feature can discriminate between walking and climbing stairs.
10. The Intensity of the movement (InMo) [11], which is the mean first derivative of the raw acceleration data, $InMo_t^{v \in \{x,y,z\}} = \frac{1}{w} \sum_{i=0}^{w-1} |a_{t-i}^v - a_{t-i-1}^v| / \Delta x t$.

Δx_t represents the time between samples, which can be ignored if the sampling rate is kept constant. The window size is given by the value of w .

11. Time Between Peaks [14], time in milliseconds between peaks in the sinusoidal waves associated with the frequency response of most activities (for each axis).
12. Binned Distribution [14,19]: as stated by the authors, this measure is used with sliding windows of size w . For each window the range should be calculated as *maximum* – *minimum*; then, the range is divided in 10 equal size bins; finally, record what fraction of the w values fell within each of the bins. This approach is called within this study Relative Binned Distribution (RBD). In this study, we also proposed the absolute binned distribution (ABD) that is calculated using the lower and upper acceleration values as the range to be divided in bins.

In many of the solutions sliding windows with or without shifting are proposed; the typical window size converges to the samples within the period of 2 seconds. Features are typically normalized to 0-mean 1-standard deviation and/or scaled to the interval [0, 1] before further preprocessing. Using frequency-derived features employing FFT or similar over long time-windows have been found more suitable for long duration, quasi-periodic signals like walking, cycling or brushing teeth. Otherwise, when classifying shorter duration and non-periodic activities, transitions or a short sequence of steps, then the time-domain representation has been found better [3]. The problem of finding the best set of features for HAR among the available transformations is the so called feature selection.

3 The Wrapper Feature Selection and HAR Method

In order to perform feature selection, this research studies a genetic algorithm driving a wrapper type FS method. The method is based on the SSGA [6], which was successfully adapted to a rather different problem [17]. Briefly, we will perform a wrapper FS method that makes use of a classifier for evaluating the individuals; each individual is a feature subset and the classifier is the GFFSM method for HAR. The method should find the feature subset that optimizes the GFFSM classifier.

3.1 GFFSM

As mentioned before, the well-known GFFSM is used [4]. This approach establishes the Finite State Machine (FSM) of the states and their transitions, the initial state machine proposed by the experts -in our case, the medical staff. Each state is considered a different label of a fuzzy variable STATE; the features are also considered fuzzy variables. A Ruspini partitioning scheme is initially proposed, and the transitions of the FSM represent the rules of the Fuzzy Rule Based System (FRBS). The GFFSM completes the learning scheme by means of a Genetic Algorithm that evolve both the fuzzy partitions and the rules in a Pittsburgh style.

As explained before, the GFFSM approach was adapted for HAR using the accelerometers on the wrists instead of having only one sensor on the back [10]. The adaptation consisted in choosing a different subset of transformations (the SMA, Δ_i and AM features) instead of those proposed in the original paper (the dorso-ventral acceleration a_i^x , the AM and the tilt of the body). This adaptation was due to the different sensor placement that makes the axes trying their orientation with the time. This section gives a brief overview of the GFFSM and also outlines the wrapper FS method we used. The fitness function is the mean absolute error (MAE), calculated as $MAE = \frac{1}{N} \frac{1}{T} \sum_{i=1}^N \sum_{j=0}^T abs(s_i[j] - s_i^*[j])$, where T is the number of examples in the data set, $s_i[t]$ and $s_i^*[t]$ are the degree of activation and the expected degree of activation, respectively, of state q_i at time $t = j$.

3.2 The SSGA-Based FS Method

An adaptation of the well known Genetic Algorithm driven wrapper feature selection algorithm called SSGA [6] was reported in [17]. This algorithm evolves the feature subset choosing the features with the lower classification error when using the KNN algorithm. In this approach, the wrapper FS also learns the GFFSM and measures the fitness of each individual acceding to Algorithms 1 and 2. The feature subsets can not be reevaluated, and if generated during the evolution then the individuals are dropped without being considered as an intermediate population individual.

Algorithm 1. IND_EVALUATION: Evaluates a feature subset

Require: $\langle I, O \rangle$ the input and output variables data sets

Require: ind the feature subset

Require: $maxFolds$ the number of cross validation runs

for each fold $k = 1$ to $maxFolds$ **do**
 generate the train and test reduced feature data set
 Run a GFFSM $\rightarrow \langle FRBS_{best}^k, mse_{best}^k \rangle$
 Keep the best $FRBS_{best}$ found
 Record the $mse_{best}^k, \forall k \rightarrow \{mse\}$

end for

Compute the average $mse_{best}^k, \forall k \rightarrow \widehat{mse}$

return $[FRBS_{best}, \widehat{mse}, \{mse\}]$

4 Evaluation of the Proposal

As a result, we have to design the experiments to validate if the above detailed method could eventually find the feature subset that optimizes the GFFSM, and the GFFSM should be not only the best model found so far by the wrapper FS method but should also enhance the previous studies. This section deals with the experimentation carried so far, that is, i) the data gathering, ii) the feature

Algorithm 2. GA⁺ Feature Selection

Require: $\langle I, O \rangle$ the input and output variables data set**Require:** N the feature subset size**Require:** $maxFolds$ the number of cross validation runsGenerate the initial population, Pop Evaluate each individual in the Pop $g \leftarrow 0$ **while** $g < G$ **do** **while** $size(Pop') < (size(Pop) - |E|)$ **do**

Generate new individuals through selection, crossover and mutation

 add valid individuals to Pop' **end while** extract the elite subpopulation $E \in Pop$ **for all** individual ind in Pop' **do** $[ind.model, ind.\widehat{mse}, \{mse\}] = IND_EVALUATION(I, O, ind, maxFolds)$ **end for** $Pop = \{E \cup Pop'\}$ sort Pop and $g++$ **end while** $FS \leftarrow Pop[0]$ $[model, mse] \leftarrow$ corresponding model and MSE**return** $[FS, model, mse]$

domain, iii) running the feature selection method and, finally, iv) comparing the obtained results and further discussion. Nevertheless, the FS approach could not be feasible in its original form and some adaptations were required. As far as the computation cost is extremely high so the time spent in each individual should be reduced. First of all, the GFFSM evolution was analyzed and it was found that the different runs within the cross validation similarly evolved and finally converged. This fact allows us to introduce some simplifications and reductions: on the one hand, the total number of individuals in a run was highly reduced for the FS evaluation. That is, during the FS evolution, the GFFSM genetic parameters were relaxed wrt those in the original paper [4]. Nevertheless, the best individual found after the FS would be trained in the same conditions in order to compare with previous studies. Moreover, the error stop condition was fixed to 0.02, allowing a certain amount of error as a compromise for reducing the computation costs. In addition, as there were no big differences between the cross validation folds, the cross validation scheme is reduced to a train-test data set. That is, five random folds were used for training and the remaining folds were kept for validation. All these simplifications would lead to obtain higher errors and perhaps the results would eventually be slightly biased; however, it is the simplest way to allow FS with a compromise between accuracy and completeness. As mentioned before, the best individual found so far would be allowed to have the complete 10-fold cross validation with the same genetic parameters than in previous studies allowing us the comparisons.

4.1 Data Gathering

To test this prototype a well-known stroke patients rehabilitation test (for short, SRT) [13] will be carried out. Two bracelets will be given to a subject, each one with a tri-axial accelerometer with sampling frequency 16 Hz. Firstly, ten runs will be registered for a normal subject. All the data will be segmented and classified according to the activity the subject is owe to do. The data for these runs will be used for training and testing the HAR in a leave-one-folder-out manner, in order to obtain statistics results. Only the data from the dominant wrist will be considered; consequently, a time series was gathered and the whole set of transformations explained in the next subsection.

4.2 The Initial Feature Subset

Using the whole set of the transformations presented in Sect. 2 leads to a rather high feature domain: each transformation induces three new variables due to the raw acceleration, but three more with the BA. Moreover, we can also calculate two more input features by means of aggregating the raw acceleration data or the BA for the three axis. Consequently, for each transformation we can calculate at most 8 new features. Instead of using the whole set of transformations, we performed first an analysis of the data and ranked the features using both the Mutual Information and the Information Correlation Coefficient [16]; previously, the features were scaled to the interval $[0, 1]$. In Fig. 1 the values of ICC for each feature in decreasing order is depicted, and the 20 features with higher ICC values were chosen.

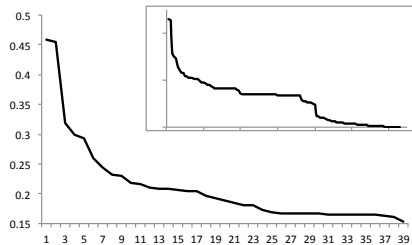


Fig. 1. Evolution of the ICC in decreasing order of the ICC value. The smaller image is the ICC for all the features.

4.3 Feature Evaluation

The parameters used for the wrapper FS GA were 30 generations with 26 individuals in the population. The one point crossover operator executed with probability 0.8, while mutation's probability was 0.02. Moreover, the GA parameters employed for the GFFSM were a population size of 76 individuals

in the population, crossover probability 0.8, the α -crossover parameter is set to 0.3, the mutation probability 0.02, the maximum number of generations set to 50, stop error condition set to 0.02 and the stop criteria of the maximum number of generations with MAE unchanged fixed to 25. Finally, with the best performance feature subset the whole GFFSM method was carried out with the original parameters (100 individuals and 200 generations). This final model was used for comparison purposes and it is referred as WRAPPER. The features finally chosen were the SMA, the Sensor Vibration and the absolute Binned mean for the X axis. In order to compare the obtained results, the GFFSM method adapted in [10] -which is called ORIG_ADAPT- and the method with several modifications for decreasing the restrictions in the crossover operators were carried out -which is called GA_MODIF-. The GA parameters in both cases were those detailed in the previous paragraph. In the second comparison method the crossover were carried out interchanging the rules at any available point instead of within each variable, thus it is expected that this method would eventually need more generations to converge. In this case, 300 generations were allowed.

4.4 Results and Further Discussion

The obtained results are depicted in Table 1 and in the box plot and MAE evolution from Fig. 2. Both models ORIG_ADAPT and WRAPPER converged, while the GA_MODIF was still evolving. Clearly, the FS method does not perform as well as expected due to the restrictions fixed on behalf of the computation cost reduction. Nevertheless, the model obtained from the chosen features undoubtedly outperforms the original method. Interesting enough, the spread of the individuals among the different cross-validation runs is kept rather small. On the other hand, the GA_MODIF results give us the clue that the learning method presented in [4] can still be optimized, perhaps by introducing learning in a Michigan style or by introducing fuzzy evaluations of the error. Finally, the selection of the best feature subset continues being a challenge that should be solved, and evaluation of other ranking and transforming measures can be introduced to choose the most suitable variables for HAR.

Table 1. Results from the different configurations and feature subsets. See the text for acronyms' definition.

Method	Train				Test			
	Best	Mean	Median	Std	Best	Mean	Median	Std
ORIG_ADAPT	0.0299	0.0361	0.0368	0.0039	0.0453	0.0802	0.0730	0.0371
GA_MODIF	0.0213	0.0293	0.0311	0.0066	0.0114	0.0391	0.0306	0.0292
WRAPPER	0.0281	0.0331	0.0325	0.0029	0.0218	0.0365	0.0343	0.0098

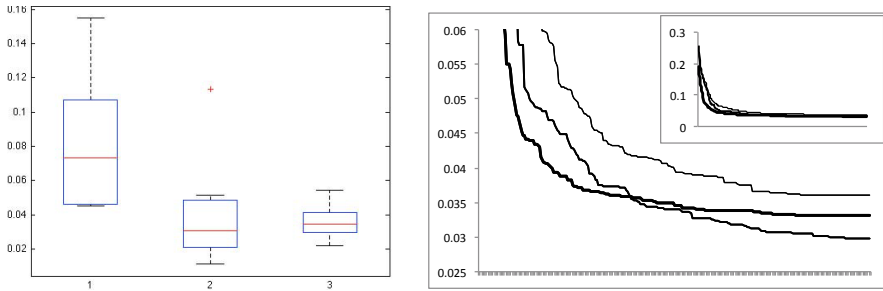


Fig. 2. Left part: Boxplot of the best individual found after the ten cross-validation runs. Right part: mean evolution of the MAE with the generations. ORIG_ADAPT, GA_MODIF and WRAPPER are marked with 1/thinner line, 2/mid-thick line and 3/thicker line on the boxplot/figure on the right, respectively. The methods with less than 300 generations appear with the best value constant filling the empty generations.

5 Conclusions and Future Work

In this study, the Genetic algorithm evolved Fuzzy Finite State Machine presented in [4] and adapted for using an accelerometer on the dominant wrist [10] is analyzed for searching the best feature subset. A wrapper FS method has been used but its performance is penalized due to the fixed computational restrictions. Nevertheless, it could be interesting to relax such restrictions as the spread obtained is the best in the comparison. Moreover, the relevance of the genetic operators and parameters is also outlined as a modified model with different crossover and higher number of generations also outperform the original method. Future work includes i) considering more computing resources to the wrapper FS method, ii) performing a more complex analysis of the genetic parameters and operators for improving the GFFSM, iii) including Michigan style solutions for learning the FSM and iv) analyzing the feature domain with more powerful methods to find out which features have more information concerning the current human activity. It is worth mentioning that this project is involved in the early stroke diagnosis using electronic intelligent devices.

Acknowledgments. This research has been supported through the Spanish Ministry of Science and Innovation TIN2011-24302 project.

References

1. Adams, H.P., del Zoppo, G., Alberts, M.J., Bhatt, D.L., Brass, L., Furlan, A., Grubb, R.L., Higashida, R.T., Jauch, E.C., Kidwell, C., Lyden, P.D., Morgenstern, L.B., Qureshi, A.I., Rosenwasser, R.H., Scott, P.A., Wijdicks, E.F.: Guidelines for the early management of adults with ischemic stroke. *Stroke* 38, 1655–1711 (2007)
2. Adams, R.D.: Principles of Neurology, 6th edn. McGraw Hill (1997)

3. Allen, F.R., Ambikairajah, E., Lovell, N.H., Celler, B.G.: Classification of a known sequence of motions and postures from accelerometry data using adapted gaussian mixture models. *Physiological Measurement* 27, 935–951 (2006)
4. Álvarez-Álvarez, A., Triviño, G., Cordon, O.: Body posture recognition by means of a genetic fuzzy finite state machine. In: *IEEE 5th International Workshop on Genetic and Evolutionary Fuzzy Systems, GEFS*, pp. 60–65 (2011)
5. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. In: Ferscha, A., Mattern, F. (eds.) *PERVASIVE 2004. LNCS*, vol. 3001, pp. 1–17. Springer, Heidelberg (2004)
6. Casillas, J., Cordon, O., del Jesus, M., Herrera, F.: Genetic feature selection in a fuzzy rule-based classification system learning process. *Information Sciences* 136(1–4), 135–157 (2001)
7. Chen, Y.P., Yang, J.Y., Liou, S.N., Lee, G.Y., Wang, J.S.: Online classifier construction algorithm for human activity detection using a tri-axial accelerometer. *Applied Mathematics and Computation* 205(2), 849–860 (2008)
8. Dromerick, A., Khader, S.A.: Medical complications during stroke rehabilitation. *Advances in Neurology* 92, 409–413 (2003)
9. Duarte, E., Alonso, B., Fernández, M., Fernández, J., Flórez, M., García-Montes, I., Gentil, J., Hernández, L., Juan, F., Palomino, J., Vidal, J., Viosca, E., Aguilar, J., Bernabeu, M., Bori, I., Carrión, F., Déniz, A., Díaz, I., Fernández, E., Forastero, P., Iñigo, V., Junyent, J., Lizarraga, N., de Munain, L.L., Máñez, I., Miguéns, X., Sánchez, I., Soler, A.: Stroke rehabilitation: Care model. *Rehabilitación* 44(1), 60–68 (2010)
10. González, S., Villar, J.R., Sedano, J., Chira, C.: A preliminary study on early diagnosis of illnesses based on activity disturbances. In: Omatu, S., Neves, J., Rodriguez, J.M.C., Paz Santana, J.F., Gonzalez, S.R. (eds.) *Distrib. Computing & Artificial Intelligence. AISC*, vol. 217, pp. 521–527. Springer, Heidelberg (2013)
11. Gyórfi, N., Fábán, Á., Hományi, G.: An activity recognition system for mobile phones. *Mobile Networks and Applications* 14, 82–91 (2009)
12. Hogdson, C.: To fast or not to fast. *Stroke* 38, 2631–2632 (2007)
13. Hollands, K.: Whole body coordination during turning while walking in stroke survivors. Ph.D. thesis, School of Health and Population Sciences. Ph.D. thesis, University of Birmingham (2010)
14. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter* 12(2), 74–82 (2010)
15. Murray, M.P., Drought, A.B., Kory, R.C.: Walking patterns of normal men. *Journal of Bone and Joint Surgery* 46(2), 335–360 (1964)
16. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Learning* 27(8), 1226–1238 (2005)
17. Villar, J.R., González, S., Sedano, J., Corchado, E., Puigpinós, L., de Ciurana, J.: Meta-heuristic improvements applied for steel sheet incremental cold shaping. *Memetic Computing* 4(4), 249–261 (2012)
18. Wang, S., Yang, J., Chen, N., Chen, X., Zhang, Q.: Human activity recognition with user-free accelerometers in the sensor networks. In: *Proceedings of the International Conference on Neural Networks and Brain, ICNN&B 2005*, vol. 2, pp. 1212–1217. *IEEE Conference Publications* (2005)
19. Yang, J.Y., Wang, J.S., Chen, Y.P.: Using acceleration measurements for activity recognition: an effective learning algorithm for constructing neural networks. *Pattern Recognition Letters* 29, 2213–2220 (2008)

Using a Hybrid Cellular Automata Topology and Neighborhood in Rule Discovery

Anca Andreica and Camelia Chira

Department of Computer Science
Centre for the Study of Complexity
Babes-Bolyai University
Cluj-Napoca, Romania
{anca,cchira}@cs.ubbcluj.ro

Abstract. Cellular Automata are important tools in the study of complex interactions and analysis of emergent behaviour. They have the ability to generate highly complex behaviour starting from a simple initial configuration and set of update rules. Finding rules that exhibit a high degree of self-organization is a challenging task of major importance in the study of complex systems. In this paper, we propose a new cellular automaton (CA) topology and neighbourhood that can be used in the discovery of rules that trigger coordinated global information processing. In the introduced approach, the state of a cell changes according to the cell itself, the cells in the local neighborhood as well as some fixed long-distance cells. The proposed topology is engaged to detect new rules using an evolutionary search algorithm for the well-known density classification task. Experiments are performed for the one-dimensional binary-state CA and results indicate a good performance of the rules evolved by the proposed approach.

Keywords: Hybrid Neighborhood, Cellular Automata Topology, Rule Evolution, Density Classification Task.

1 Introduction

Cellular Automata (CAs) are decentralized structures of simple and locally interacting elements (cells) that evolve following a set of rules [23]. Programming CAs is not an easy task, particularly when the desired computation requires global coordination. CAs provide an idealized environment for studying how (simulated) evolution can develop systems characterized by emergent computation where a global, coordinated behavior results from the local interaction of simple components [14]. However, the discovery of rules exhibiting a high degree of global self-organization is not easily achieved since coordinated global information processing must rise from the interactions of components with local information and communication.

The one-dimensional binary-state CA capable of performing computational tasks has been extensively studied in the literature [9,19,15,16,1]. One of the

widely studied CA problems is the density classification task (DCT). Most existing studies [5,2,14,8,15,18,13,10,16] focus on developing algorithms able to find high performant rules for 1D CAs with fixed neighborhood size.

In this paper, we investigate a new neighborhood structure for lattice-based CAs in which the state change of a cell is allowed to be influenced by long-distance cells. Besides the cell itself and the cells in the local neighborhood, fixed distant cells contribute to the way in which the cell state is changed over time. This hybrid neighborhood has a fixed structure combining local and, to a certain extent, global information. The rule able to correctly classify random initial configurations is difficult to find because the search takes into account only local information and very limited information coming from long-distant connecting cells. In this approach, the rule can vary depending on the states of the neighborhood and is always applied to the same set of cells. Compared to the standard neighborhood in lattice-based CAs, the proposed approach introduces a connection of each cell with a limited fixed number of long-distant cells while the neighborhood size remains fixed. In both cases, the search focuses on the best performing rule in connection with a fixed neighborhood. While this neighborhood structure is fixed itself in the standard lattice model, the proposed approach allows a variable neighborhood structure depending on the cells selected as long-distance influence. The similarities between the proposed hybrid neighborhood topology and network-based CAs lie only at the level of cell neighborhood to a certain extent in the sense that both approaches allow a variable neighborhood (the former by including a different cell structure in the neighborhood of each cell, and the latter by varying both the content and the size of the neighborhood). However, the focus in network-based CAs is to detect the best network topology in connection with a fixed rule while, in the proposed approach, the search process focuses on finding best rule in connection with a hybrid neighborhood.

An evolutionary algorithm is developed to search for the best performing rule for DCT based on the proposed hybrid neighborhood structure. Computational experiments are performed for one-dimensional CAs and results emphasize a better rule performance compared to regular lattice topologies with a standard neighborhood.

The rest of the paper is structured as follows: section 2 presents the density classification task and the relevant lattice and network-based CA topologies, section 3 describes the proposed CA hybrid neighborhood topology, section 4 presents the evolutionary framework proposed for solving DCT, section 5 includes the computational experiments and results, and section 6 contains the conclusions of the paper and direction for future work.

2 Density Classification Task and CA Topologies

The aim of DCT is to find a binary one-dimensional CA able to classify the density of 1s (denoted by ρ_0) in the initial configuration. If $\rho_0 > 0.5$ (1 is dominant in the initial configuration) then the CA must reach a fixed point configuration

of 1s otherwise it must reach a fixed-point configuration of 0s within a certain number of timesteps. Most studies consider the case $N = 149$ (which means that the majority is always defined) and neighborhood size of 7 (the radius of CA is $r = 3$). The CA lattice starts with a given binary string called the initial configuration (IC). After a maximum number of iterations (usually set as twice the size of CA), the CA will reach a certain configuration. If this is formed of homogeneous states of all 1s or 0s, it means that 1, respectively 0, is dominant in the initial configuration. Otherwise, the CA makes by definition a misclassification [18]. It has been shown that there is no rule that can correctly classify all possible ICs [12].

The performance of a rule measures the classification accuracy of a CA based on the fraction of correct classifications over ICs selected from an unbiased distribution (ρ_0 is centered around 0.5). DCT is a challenging problem extensively studied due to its simple description and potential to generate a variety of complex behaviors. As already mentioned, most existing studies [5,2,14,8,15,18,13,10,16] focus on the detection of rules for 1D CAs with fixed neighborhood. Each iteration, the change of a cell state is in function of the cell itself and r local cells neighbors on each side. In this case, the neighbourhood size is fixed to $2 * r + 1$ and the method searches for the best rule able to correctly classify potentially any IC. Network-based CAs have also been investigated in the context of DCT [21,22,3,4,20]. In this case, the topology of the CA is given by a general graph and the neighborhood of each cell varies with the number of connecting nodes in the graph while the rule remains fixed (i.e. the majority rule). For network-based CAs, algorithms are designed to search for the graph topology that triggers the best neighborhood to be used in connection with a fixed rule for the DCT.

The CA topology and neighbourhood structure used for a cell in applying the rule are crucial elements in the process of rule discovery and impact directly the rule performance. There are two existing CA topologies used for the 1D CA in connection with DCT: the regular lattice with fixed-size neighbourhood and the network topology with neighbourhood induced by the network structure.

Usually, a one-dimensional lattice of N two-state cells is used for representing the CA. The state of each cell changes according to a function depending on the current states in the neighborhood (the cell itself and its r neighbors on both sides of the cell, where r represents the radius of the CA). The regular lattice topology is engaged in most studies tackling DCT for 1D binary-state CA [17,11,6,5,2,14,8,15,18,13,10,16].

A few studies [21,22,3,4,20,7] consider an extension of the CA concept in which the cells can be connected in any way while the rule is the same for all cells. In this approach, the topological structure of CAs refers to general graphs.

3 Proposed Hybrid Topology and Neighborhood

The two topologies mentioned in the previous section are often compared even if they use different rules for determining the next cell state. Indeed, in the case of network topologies, due to the variable number of neighbors, the rule can not be

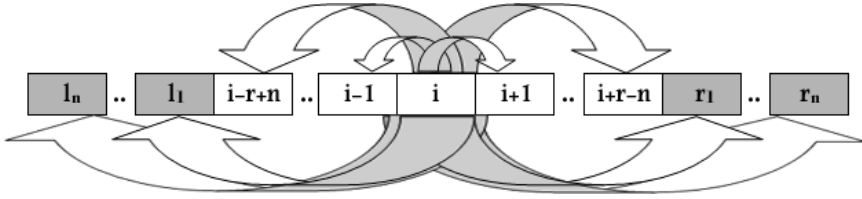


Fig. 1. Neighborhood of node i in the new proposed topology

the same as in the case of regular lattices. Therefore, the advantage that seems to be brought by the good performance obtained when using network topologies could be also motivated by the different rule that is applied in this case.

In this paper, we propose a new hybrid topology and a mixed induced neighborhood that keeps invariable the number of neighbors, this way allowing a fairer comparison with regular lattice topologies.

The neighborhood of a node is given by the radius r . Each node has r neighbors on the left hand side and r neighbors on the right hand side, which gives a neighborhood of $2 * r + 1$, because we also consider the node itself. In order to create the new topology of radius r , we start with a regular ring lattice of radius $r - n$. The other $2 * n$ nodes that node i is connected to, are long distance neighbors. They are randomly chosen from the rest of the nodes, but following some rules that ensure the equilibrium of the neighborhood. This means that i always remains the central node of the neighborhood and the distance between node i and the long distance neighbors places half of them (n nodes) on the left hand side and the other half (n nodes) on the right hand side of i . Figure 1 depicts the neighborhood of node i in the new proposed topology.

For example, for $r = 3$ and $n = 1$, node i is connected to nodes $i - 2$ and $i - 1$ on one hand, and with nodes $i + 1$ and $i + 2$ on the other hand, and is connected to two long distance nodes, one on the left hand side of i (l_1) and the other one on the right hand side of node i (r_1).

The rules that govern the choice of the far distance neighbors are given in what follows.

Let us denote by *left* the leftmost neighbor and by *right* the rightmost neighbor from the regular ring lattice of radius $r - n$. Let us denote by N the number of nodes.

Each long distance neighbor $l_k, k = \{1, \dots, n\}$, is generated from half of the possible nodes, the closest ones to the node *left*. Therefore, we have the following two cases:

$$(1) \text{left} - \left(\frac{N-1}{2} - (r - n)\right) \geq 0$$

which means that all possible left hand side long distance neighbors are above 0. In this case, one long distance neighbor $l_k, k = \{1, \dots, n\}$ is randomly generated from the set:

$$\{\text{left} - \left(\frac{N-1}{2} - (r - n)\right), \dots, \text{left} - 1\}$$

The second case is:

$$(2) \text{left} - \left(\frac{N-1}{2} - (r - n)\right) < 0$$

which means that some of the possible left long distance neighbors are above 0, and some of them are below N . We have to take this situation into account because we deal with a ring lattice (which means that, for example, the closest neighbors of 0 in a regular ring lattice are 1 and $N - 1$).

Therefore, the long distance neighbor $l_k, k = \{1, \dots, n\}$ could be randomly generated either from the set:

$$\{0, \dots, left - 1\}$$

or from the set

$$\{\frac{N-1}{2} + (r - n) + left + 1, \dots, N - 1\}.$$

In order to better illustrate the above formulas, let us consider the following example:

$$N = 149, r = 3, n = 1,$$

which means that we have a cellular automata with 149 cells, the radius is 3 so the neighborhood is 7, we take 2 neighbors on each side from the regular ring lattice of radius 2 and we generate another 2 lost distance neighbors, one for each side of a cell.

(a) for $left = 75$, we have case (1) because $3 \geq 0$. We therefore generate a left long distance neighbor from the set $\{3, \dots, 74\}$.

(b) for $left = 30$, we have case (2) because $-42 < 0$. We therefore generate a left long distance neighbor either from set $\{0, \dots, 29\}$ or from set $\{107, 148\}$.

Each long distance neighbor $r_k, k = \{1, \dots, n\}$ is generated from half of the possible nodes, the closest ones to the node $right$. Similar to the way we generate left long distance neighbors, we also have two possible cases when generating long distance neighbors for the right hand side of a node:

$$(3) \quad right + (\frac{N-1}{2} - (r - n)) < N$$

case in which all nodes will be generated from the set:

$$\{right + 1, \dots, right + \frac{N-1}{2} - (r - n)\},$$

because all possible nodes are smaller than N .

$$(4) \quad right + (\frac{N-1}{2} - (r - n)) \geq N$$

which means that possible values for $r_k, k = \{1, \dots, n\}$ could be either taken from the set

$$\{right + 1, \dots, N - 1\},$$

or from the set

$$\{0, \dots, \frac{N-1}{2} - (r - n) - N + right\}.$$

Let us take one example for the right long distance neighbors too. For the same $N = 149, r = 3, n = 1$, let us consider:

(c) $right = 35$, which takes us to case (3) because $107 < 149$. We therefore generate a right long distance neighbor from set $\{36, \dots, 107\}$.

(d) $right = 100$, which means that we are in case (4) because $172 > 149$. This means that we generate the long distance neighbor either from set $\{101, \dots, 148\}$ or from set $\{0, \dots, 23\}$.

By using this approach we obtain a new topology which resembles a network topology but it is still very close to a regular ring lattice, which allows us to consider the same majority understanding as in the classical case of regular ring lattice topology.

4 Evolutionary Search Algorithm

A simple evolutionary framework has been set up to detect rules for the DCT. A potential solution of the problem is encoded as a one-dimensional array of bits of size $2^{2r+1} = 128$ (because we have considered the radius as having the value $r = 3$) and represents a rule table for the cellular automaton. The initial population is randomly generated.

The potential solutions are evaluated by means of a real-valued fitness function $f : X \rightarrow [0, 1]$, where X denotes the search space of the problem. As stated before, $|X| = 2^{128}$. The fitness function represents the fraction of correct classification over 100 randomly generated initial configurations. A relative fitness is used, as the set of initial configurations is generated anew for each generation of the algorithm. This way, solutions with high fitness in one generation and which survive in the next generation will be evaluated again using another set of 100 initial configurations.

Every set of 100 initial configurations was generated so that their densities are uniformly distributed over $[0,1]$. It is important to underline the difference between the fitness of a rule and the performance of a rule. While the fitness is evaluated by using 100 uniformly distributed initial configurations, the performance of a rule is computed as the fraction of correct classifications for 10^4 randomly generated initial configurations. The initial configurations are generated in such a way that each cell has the same probability $\frac{1}{2}$ of being 0 or 1. This means that the density of 1s will be around $\frac{1}{2}$ for most of the initial configurations and these are actually the most difficult cases to correctly classify. The CA is iterated until it reaches a fixed-point configuration of 1s or 0s but for no more than $M \approx 2N$ time steps.

The individual resulted after each recombination will be mutated at exactly two randomly chosen positions. A weak mutation is considered, the probability of obtaining a different value for the chosen position being equal to the probability of obtaining the very same value. The algorithm is applied for 100 generations with a population size of 100, roulette selection, one point crossover with probability of 0.8, weak mutation with probability 0.2 and elite size of 10%.

It should be noted that the same evolutionary algorithm with the same parameters is used to evolve rules for both regular lattice and proposed topology to allow a direct comparison of results.

5 Computational Experiments and Results

Experiments focus on the most frequently studied version of the DCT problem: the one-dimensional binary-state CA of size 149 based on the radius of 3. In the case of regular lattice topology, this means that each cell is connected to 3 neighbors from both sides while for the proposed topology we set $r = 3$ and $n = 1$ (i.e. 2 local neighbours on each side and 1 distant neighbor). In both cases, the neighborhood size is 7 cells which leads to a rule size of $2^7 = 128$. The number of all possible rules is 2^{128} which makes an exhaustive evaluation of all this rules unfeasible.

Table 1. Performance obtained in 10 runs for both RegCA and HybCA

	RegCA	HybCA
Best	0.6436	0.7902
Average	0.62852	0.74847
Std Dev	0.012619192	0.034297815
T-test p-value	0.000005	

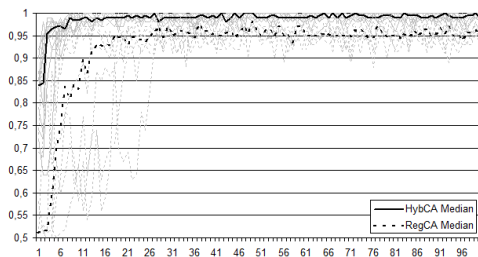


Fig. 2. Evolution of fitness in 10 runs of RegCA and 10 runs of HybCA

In order to test the efficiency of the proposed topology, we perform a comparison between CAs with regular ring lattice topology (RegCA) and CAs with new proposed hybrid topology (HybCA) using the evolutionary algorithm described in the previous subsection.

Firstly we test the performance obtained by a regular CA compared with a CA based on the new proposed topology. Obtained results are presented in Table 1.

The results obtained after 10 runs of the algorithm for RegCA and HybCA, presented in Table 1, are compared using the paired t-test with a 95% confidence interval. The p-value is significantly smaller than 0.05, which indicates that the mean performances obtained when using the proposed neighborhood are notably better than those obtained when using a regular lattice topology. The best performance obtained in 10 runs of the algorithm for a regular lattice CA is considerably improved from 0.64 to 0.79 when changing the topology by replacing two neighbors with 2 long distance nodes.

The evolution of the fitness for all 10 runs of both RegCA and HybCA is depicted in Figure 2. The median values are bold in the chart. In the case of HybCA, the fitness is most of the time higher compared to RegCA, being closest to the maximum value of 1.

The CA diagram obtained for one of the best rules given by HybCA is depicted in Figure 3.

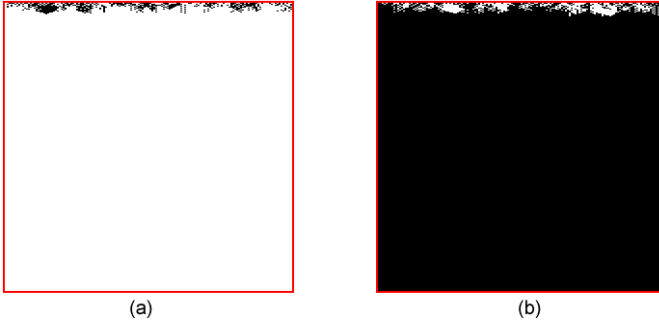


Fig. 3. CA diagram for a rule with performance 0.7715 obtained by HybCA. $\rho_0 < 0.5$ in figure (a) and $\rho_0 > 0.5$ in figure (b).

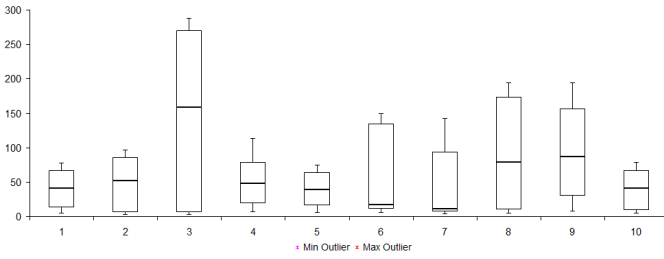


Fig. 4. Number of iterations needed to correctly classify the initial configurations for RegCA, for each of 10 runs

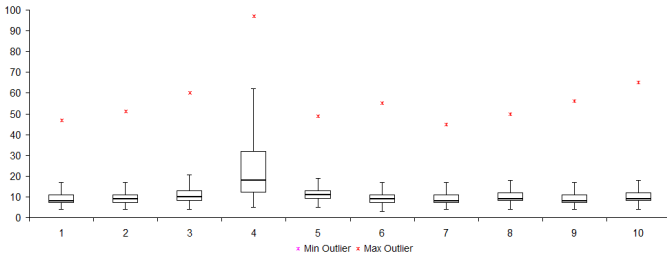


Fig. 5. Number of iterations needed to correctly classify the initial configurations for HybCA, for each of 10 runs

Another comparison that clearly shows the efficiency of the proposed neighborhood is the number of iterations needed to correctly classify the initial configurations (for example, for a performance of 0.71, there are 7100 correctly classified initial configurations and therefore 7100 observations of how many iterations were needed for those correctly classified initial configurations). Obtained results for 10 different runs are depicted in Figures 4 and 5. It can be easily observed that in the case of RegCA the number of needed iterations is

considerably higher compared to HybCA. This means that the convergence is significantly accelerated when using the proposed topology.

6 Conclusions and Future Work

A new hybrid neighborhood structure for lattice-based CAs has been proposed and investigated in connection with the density classification task. The state of a cell changes in function of the cell itself, some cells from the local neighborhood and a fixed number of long-distance cells. This forms a hybrid neighborhood topology for which the best performing rules in connection with DCT are evolved in a search process. Computational experiments emphasize that the rules detected by an evolutionary algorithm have a good performance shown to improve that of rules evolved for the standard neighborhood used in lattice CAs.

Future work focuses on extended experiments for various neighborhood sizes and the trade-off between the number of local cells and long-distant cells in the considered structure. Moreover, other rules will be investigated in connection with the proposed hybrid neighborhood and CA topology.

Acknowledgment. This research is supported by Grant PN II TE 320, Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCSIS, Romania.

References

1. Chira, C., Gog, A., Lung, R.I., Iclanzan, D.: Complex Systems and Cellular Automata Models in the Study of Complexity. *Studia Informatica Series LV(4)*, 33–49 (2010)
2. Crutchfield, J.P., Mitchell, M.: The evolution of emergent computation. *Proceedings of the National Academy of Sciences, USA* 92(23), 10742–10746 (1995)
3. Darabos, C., Giacobini, M., Tomassini, M.: Performance and Robustness of Cellular Automata Computation on Irregular Networks. *Advances in Complex Systems* 10, 85–110 (2007)
4. Darabos, C., Tomassini, M., Di Cunto, F., Provero, P., Moore, J.H., Giacobini, M.: Toward robust network based complex systems: from evolutionary cellular automata to biological models. *Intelligenza Artificiale* 5(1), 37–47 (2011)
5. Das, R., Mitchell, M., Crutchfield, J.P.: A genetic algorithm discovers particle-based computation in cellular automata. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) *PPSN 1994. LNCS*, vol. 866, pp. 344–353. Springer, Heidelberg (1994)
6. Ferreira, C.: Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. *Complex Systems* 13(2), 87–129 (2001)
7. Gog, A., Chira, C.: Dynamics of Networks Evolved for Cellular Automata Computation. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) *HAIS 2012, Part II. LNCS*, vol. 7209, pp. 359–368. Springer, Heidelberg (2012)
8. Hordijk, W., Crutchfield, J.P., Mitchell, M.: Mechanisms of Emergent Computation in Cellular Automata. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) *PPSN 1998. LNCS*, vol. 1498, pp. 613–622. Springer, Heidelberg (1998)

9. Juille, H., Pollack, J.B.: Coevolving the ‘ideal’ trainer: Application to the discovery of cellular automata rules. *Genetic Programming 1998: Proceedings of the Third Annual Conference* (1998)
10. Juille, H., Pollack, J.B.: Coevolutionary learning and the design of complex systems. *Advances in Complex Systems* 2(4), 371–394 (2000)
11. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge (1992)
12. Land, M., Belew, R.K.: No perfect two-state cellular automata for density classification exists. *Physical Review Letters* 74(25), 5148–5150 (1995)
13. Marques-Pita, M., Mitchell, M., Rocha, L.M.: The role of conceptual structure in designing cellular automata to perform collective computation. In: Calude, C.S., Costa, J.F., Freund, R., Oswald, M., Rozenberg, G. (eds.) *UC 2008. LNCS*, vol. 5204, pp. 146–163. Springer, Heidelberg (2008)
14. Mitchell, M., Crutchfield, J.P., Das, R.: Evolving cellular automata with genetic algorithms: A review of recent work. In: *Proceedings of the First International Conference on Evolutionary Computation and Its Applications, EvCA 1996*. Russian Academy of Sciences (1996)
15. Mitchell, M., Thomure, M.D., Williams, N.L.: The role of space in the Success of Coevolutionary Learning. In: *Proceedings of ALIFE X - The Tenth International Conference on the Simulation and Synthesis of Living Systems* (2006)
16. Oliveira, G.M.B., Martins, L.G.A., de Carvalho, L.B., Fynn, E.: Some investigations about synchronization and density classification tasks in one-dimensional and two-dimensional cellular automata rule spaces. *Electron. Notes Theor. Comput. Sci.* 252, 121–142 (2009)
17. Packard, N.H.: Adaptation toward the edge of chaos. In: *Dynamic Patterns in Complex Systems*, pp. 293–301. World Scientific (1988)
18. Pagie, L., Mitchell, M.: A comparison of evolutionary and coevolutionary search. *Int. J. Comput. Intell. Appl.* 2(1), 53–69 (2002)
19. Tomassini, M., Venzi, M.: Evolution of Asynchronous Cellular Automata for the Density Task. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) *PPSN 2002. LNCS*, vol. 2439, pp. 934–943. Springer, Heidelberg (2002)
20. Tomassini, M., Giacobini, M., Darabos, C.: Evolution and dynamics of small-world cellular automata. *Complex Systems* 15, 261–284 (2005)
21. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘smallworld’ networks. *Nature* 393, 440–442 (1998)
22. Watts, D.J.: *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton University Press, Princeton (1999)
23. Wolfram, S., *Theory and Applications of Cellular Automata*. Advanced series on complex systems. World Scientific Publishing (1986)

An Extension of the FURIA Classification Algorithm to Low Quality Data

Ana Maria Palacios¹, Luciano Sanchez², and Ines Couso^{2,*}

¹ Computer Science Department
Granada University
18071 Granada, Spain
palacios@decsai.ugr.es

² Computer Science Department
Oviedo University
33204 Gijon, Spain
{luciano,couso}@uniovi.es

Abstract. The classification algorithm FURIA (Fuzzy Unordered Rule Induction Algorithm) is extended in this paper to low quality data. An epistemic view of fuzzy memberships is adopted for modeling the incomplete knowledge about training and test sets. The proposed algorithm is validated in different real-world problems and compared to alternative fuzzy rule-based classifiers in both their linguistic understandability and the accuracy of the results. Statistical tests for vague data are used to show that the new algorithm has a competitive edge over previous approaches, especially in some high dimensional problems.

Keywords: Low Quality Data, FURIA, Fuzzy Rule Based Classifiers.

1 Introduction

The expression “machine learning tasks with low quality data” (LQD) alludes to those problems where some properties of the elements of the training and test sets cannot be precisely observed, and the probability distribution of the observation errors is also partially unknown. The use of possibility distributions allows to model this kind of incomplete knowledge in many practical cases, including censored, missing, interval or fuzzy-valued data with an epistemic interpretation, compound data, linguistic terms, etc. [14]

In this respect, there only exist a few algorithms that can learn rule-based classifiers from possibilistic data. For instance, GCCL-LQD [11] and Boosting-LQD [13] combine genetic algorithms and fuzzy-valued fitness functions to produce linguistic rules from vague information. However, these extensions are not based on the most recent classification algorithms in the literature. It makes sense to check whether there is room for improvement, by extending the newest

* This work was supported by the Spanish Ministerio de Economía y Competitividad under Project TIN2011-24302, including funding from the European Regional Development Fund.

rule induction algorithms with the same set of techniques that were used for generalizing GCCL and Adaboost to LQD. Because of this, in this contribution a state-of-the-art classification algorithm, named “FURIA” (Fuzzy Unordered Rule Induction Algorithm) will be extended to low quality data, understood as datasets comprising tuples of fuzzy sets, whose α -cuts are, in turn, random sets which contain the value of each unknown crisp feature with a probability greater or equal than $1-\alpha$ [4]. An experimental study is therefore provided, where GCCL, Boosting and the proposed extension of FURIA are compared for some real-world problems involving vague data. Extensions to LQD of some statistical tests commonly used in machine learning experimental designs [5] will be used for judging the relevance of the differences [10].

This paper is organized as follows. Section 2 briefly introduces the FURIA algorithm and remarks the parts that are more relevant for this proposal. In Section 3, the changes effected to this algorithm are detailed. In Section 4, numerical results are given. Concluding remarks and future work are discussed in Section 5.

2 Outline of the FURIA Algorithm

Fuzzy Unordered Rules Induction Algorithm (FURIA) [7,9] is a novel fuzzy rule-based classification method extending the classical RIPPER [3]. The parts of FURIA that will be modified in Section 3 are described here for the convenience of the reader, who is referred to [7,9] and also to the source code of the software implementation provided by the authors in [8] for a full description of this algorithm.

In the following, the training set is $D \subset \mathbb{R}^k$ and instances are vectors $x = (x_1, \dots, x_k) \in D$. Each antecedent of a FURIA fuzzy classification rule is a multivariate trapezoidal fuzzy set whose membership is

$$I^F(x) = \bigoplus_{i=1, \dots, k} I_i^F(x_i) \tag{1}$$

and its core is the interval $I = I_1 \times \dots \times I_k$, where the indicator function of I_i , $i = 1, \dots, k$ is

$$I_i(x_i) = \begin{cases} 1 & \text{if } I_i^F(x_i) = 1 \\ 0 & \text{else.} \end{cases} \tag{2}$$

The parts of FURIA that will be generalized to low quality data, as mentioned, are described below:

- **Information Gain:** This criterion measures the improvement of a rule with respect to the default for the target class and is used as a stopping condition in the rule growing procedure. Let I be the core of the antecedent of the rule at hand, and let l be the target class. Then, the number of positive examples for the fuzzy classification rule r is

$$p_r = \#\{x \in I \mid \text{class}(x) = l\} \tag{3}$$

and the number of negative examples for that rule is

$$n_r = \#\{x \in I \mid \text{class}(x) \neq l\}. \quad (4)$$

The total number of positive and negative examples in the dataset are named p and n , respectively. Then, the information gain is defined as follows [8]:

$$\text{IG}_r = p_r \times \left(\log_2\left(\frac{p_r + 1}{p_r + n_r + 1}\right) - \log_2\left(\frac{p + 1}{p + n + 1}\right) \right). \quad (5)$$

- **Pruning:** Rules comprise q antecedents $\langle a_1, \dots, a_q \rangle$ combined with the AND operator. The order of the antecedents reflects their importance thus pruning a rule consists of selecting a sublist $\langle a_1, \dots, a_i \rangle$, with $i \leq q$. In order to find a suitable value for i , the following rule-value metric is computed first [8]:

$$V_r = \frac{p_r + 1}{p_r + n_r + 2} \quad (6)$$

Let the number of positive covered and negative uncovered examples of the rule, when pruned at the i -th antecedent, respectively be P_i and N_i :

$$P_i = \#\{x \mid x \text{ is covered by } \langle a_1, \dots, a_i \rangle \wedge \text{class}(x) = l\} \quad (7)$$

$$N_i = \#\{x \mid x \text{ is not covered by } \langle a_1, \dots, a_i \rangle \wedge \text{class}(x) \neq l\}. \quad (8)$$

and let be defined the value [8]

$$\text{worth}_i = \frac{P_i + N_i}{p + n}. \quad (9)$$

This value measures how likely is each antecedent to be pruned. If

$$\max_{i=1, \dots, q} \text{worth}_i > V_r, \quad (10)$$

then the term where the value of “worth_{*i*}” is maximum is selected for pruning.

- **Purity:** This value measures the quality of the fuzzification procedure and it is used for determining the support of the fuzzy sets defining the rule antecedents. Let D^i be the subset of the training data that follows:

$$D^i = \{(x_1, \dots, x_k) \mid x_j \in I_j^F(x_j) \text{ for all } j \neq i\}. \quad (11)$$

D^i is partitioned into positive and negative instances, D_+^i and D_-^i . Given the values

$$p_i = \sum_{x \in D_+^i} I_i^F(x_i) \quad (12)$$

$$n_i = \sum_{x \in D_-^i} I_i^F(x_i), \quad (13)$$

the purity of the fuzzification of the i -th attribute is [8]:

$$\text{pur}_i = \frac{p_i}{p_i + n_i} \quad (14)$$

- **Certainty Factor:** The certainty factor CF of a rule $\langle I^F, l \rangle$, for a training set D_T , is [8]:

$$CF = \frac{2 \cdot \frac{\sum_{x \in D_T, \text{class}(x)=l} p(x)}{\sum_{x \in D_T} p(x)} + \sum_{x \in D_T, \text{class}(x)=l} I^F(x)}{2 + \sum_{x \in D_T} I^F(x)} \tag{15}$$

where $p(x)$ is the weight of instance x , often 1.

- **Rule Stretching:** Rule stretching (or generalization) deals with uncovered examples (those classified by the default rule in RIPPER). The generalization procedure consists of making (preferably minimal) simplifications of the antecedents of the rules until the query instance is covered. The instance is then classified by the rule with the highest evaluation, according to the value [8]

$$STR = CF \cdot \frac{k + 1}{m + 2} \cdot I^F(x) \tag{16}$$

where k is the size of the generalized antecedent and m is the size of the entire antecedent before applying this procedure. Notice that, $\frac{k+1}{m+2}$ aims at discarding heavily pruned rules. If no stretched rule is able to cover the given example x_i , it is assigned a class based on the *a priori* distribution.

3 An Extension of FURIA to Low Quality Data

Before the expressions mentioned in the preceding section are detailed, the following definitions are needed:

- **Low Quality Datasets:** Let $(\widetilde{X}_1, Z_1) \dots (\widetilde{X}_n, Z_n)$ be a set of vague data, where n is the number of instances, $\widetilde{X}_i \in \mathcal{F}(R^d)$ and $Z_i \subset C = \{c_1, \dots, c_m\}$. It is remarked that imprecision is allowed in the output variable. Multiple labels should be understood as an indeterminacy in the class label of an object.
- **Number of Instances of a Given Class:** The number of instances of class c_j is a fuzzy number \overline{f}_{c_j} , and the same happens to the relative frequencies of the classes \overline{fr}_{c_j} .

$$\overline{f}_{c_j} = \sum_{i=1}^n \overline{\delta}_{c_j, Z_i} \tag{17}$$

where the set-valued function $\overline{\delta}$ is defined as follows:

$$\overline{\delta}_{a,A} = \{\delta_{a,b} : b \in A\} = \begin{cases} \{1\} & \{a\} = A, \\ \{0\} & a \notin A, \\ \{0, 1\} & \text{else.} \end{cases} \tag{18}$$

The relative frequency of a class takes into account the weights p_i of the instances:

$$\overline{fr}_{c_j} = \frac{\sum_{i=1}^n \overline{\delta}_{c_j, Z_i} p_i}{\sum_{i=1}^n p_i} = \frac{\sum_{i=1}^n \overline{P}_{c_j}^i}{\sum_{i=1}^n p_i} \tag{19}$$

where

$$\overline{P}_{c_j}^i = \begin{cases} \{p_i\} & c_j = Z_i, \\ \{0\} & c_j \notin Z_i, \\ \{0, p_i\} & \text{else.} \end{cases} \tag{20}$$

- **Class Being Processed:** The class being processed c_j is defined by the default number of correct classifications:

$$\overline{\text{defAccRT}}_{c_j} = \frac{1 + \overline{\text{defAccu}}_{c_j}}{1 + \sum_{i=1}^n p_i} \tag{21}$$

where

$$\overline{\text{defAccu}}_{c_j} = \sum_{i=1}^n \overline{\delta}_{c_j, Z_i} p_i = \sum_{i=1}^n \overline{P}_{c_j}^i \tag{22}$$

The stopping criterion when creating rules of class c_j is $1 - \overline{\text{defAccRT}}_{c_j} > \text{Threshold}$, whose extension is

$$\begin{aligned} P(\overline{\text{defAccRT}}_{c_j} > \text{Threshold}) &> 0.5 = \\ P([a_{c_j}, b_{c_j}] > \text{Threshold}) &> 0.5 = \\ = P([a_{c_j} - \text{Threshold}, b_{c_j} - \text{Threshold}] > 0) &> 0.5 \end{aligned} \tag{23}$$

These definitions considered, the following methods of FURIA (see Section 2) have to be altered:

- **Information Gain:** The number of positive and negative examples are imprecise. Let I be the core of the antecedent of the rule at hand, and let c_l be the target class. Then, the information gain is defined as follows:

$$\overline{IG}_r = \sum_{\tilde{X}_i \in I} \overline{P}_{c_l}^i \cdot (\log_2(\overline{\text{fstAccuRate}}_{c_l}) - \log_2(\overline{\text{defAccRT}}_{c_l})) \tag{24}$$

where

$$\overline{\text{fstAccuRate}}_{c_l} = \frac{1 + \overline{\text{fstAccu}}_{c_l}}{1 + \text{Coverfst}_{c_l}} \tag{25}$$

$$\overline{\text{fstAccu}}_{c_l} = \sum_{\tilde{X}_i \in I} \overline{P}_{c_l}^i \tag{26}$$

$$\text{Coverfst}_{c_l} = \sum_{\tilde{X}_i \in I} p_i \tag{27}$$

- **Rule Pruning:** This consists in finding the position in the antecedent list of the rule $\langle a_1, \dots, a_q \rangle$ with $i \leq q$ where the rule must be split, according to the following criteria:

1. The value defined in Eq. 6 is extended as follows:

$$\overline{V}(r) = \frac{1 + \overline{\text{defAccu}}_{c_j}}{2 + \sum_{i=1}^n p_i} \tag{28}$$

2. For each antecedent, the number of positive covered instances $\overline{\text{Pos}}_{a_m}$ and negative uncovered instances $\overline{\text{Neg}}_{a_m}$ are:

$$\overline{\text{Pos}}_{a_m} = \sum_{\tilde{x}_i \text{ is covered by } \langle a_1, \dots, a_m \rangle} \overline{P}_{c_j}^i \tag{29}$$

$$\overline{\text{Neg}}_{a_m} = \sum_{\tilde{x}_i \text{ is not covered by } \langle a_1, \dots, a_m \rangle} \overline{Pn}_{c_j}^i \tag{30}$$

where

$$\overline{Pn}_{c_j}^i = \begin{cases} \{p_i\} & (\{c_j\} \neq Z_i \text{ and } \#Z_i = 1) \text{ or } (c_j \notin Z_i), \\ \{0, p_i\} & c_j \in Z_i \text{ and } c_j \neq Z_i. \end{cases} \tag{31}$$

3. The net worth of each antecedent is

$$\overline{\text{worth}}_{a_m} = \frac{\overline{\text{Pos}}_{a_m} + \overline{\text{Neg}}_{a_m}}{\sum_{i=1}^n p_i}, \tag{32}$$

and this last value is used to decide splitting position, as shown below.

4. If $\overline{\text{worth}}_{a_m}$, with $m = 1, \dots, q$, dominates $\overline{V}(r)$, the splitting point is the m -th antecedent.
- **Purity:** Let \tilde{D}^i be the subset of the training data that follows:

$$\tilde{D}^i = \{(\tilde{X}_1, \dots, \tilde{X}_k) \mid \text{match}(\tilde{X}_j, \tilde{A}_m) \text{ for all } m \neq i\} \tag{33}$$

where

$$\text{match}(\tilde{X}_j, \tilde{A}_m) \approx \text{match}([X_j]_\alpha, \tilde{A}_m) = \{\tilde{A}_m(x) \mid x \in [X_j]_\alpha\}. \tag{34}$$

The purity of the fuzzification of the i -th antecedent is extended as explained below:

$$\overline{\text{pur}} = \frac{\overline{p}_i}{\overline{p}_i + \overline{n}_i} \tag{35}$$

where

$$\overline{p}_i = \sum_{j=1}^{\#\tilde{D}^i} (\text{match}(\tilde{X}_j, \tilde{A}_i) * \overline{P}_{c_i}^j) \tag{36}$$

$$\overline{n}_i = \sum_{j=1}^{\#\tilde{D}^i} (\text{match}(\tilde{X}_j, \tilde{A}_i) * \overline{Pn}_{c_i}^j) \tag{37}$$

- **Certainty Factor:** The certainty factor of a rule $\langle a_1, \dots, a_q \rangle$ whose consequent is c_j , for the training set \tilde{D}_T , is:

$$\overline{CF} = \frac{2 \cdot \frac{\sum_{\tilde{x}_i \in \tilde{D}_T, 1 \leq i \leq n} \overline{P}_{c_j}^i}{\sum_{\tilde{x}_i \in \tilde{D}_T, 1 \leq i \leq n} p_i} + \sum_{\tilde{x}_i \in \tilde{D}_T, 1 \leq i \leq n} \text{match}_1(\tilde{x}_i, \tilde{A}) * \overline{P}_{c_j}^i}{2 + \sum_{\tilde{x}_i \in \tilde{D}_T, 1 \leq i \leq n} \text{match}_1(\tilde{x}_i, \tilde{A}) * p_i} \tag{38}$$

where

$$\begin{aligned} \text{match}_1(\tilde{X}_j, \tilde{A}) &\approx \text{match}([X_j]_\alpha, \tilde{A}) = \\ &\{T - \text{norm}(\tilde{A}_i(x), 1 \leq i \leq q) | x \in [X_j]_\alpha\} \end{aligned} \tag{39}$$

- **Rule Stretching:** The extension of Eq. 16 is straightforward:

$$\overline{STR} = \overline{CF} \cdot \frac{k + 1}{m + 2} \cdot \text{match}(\tilde{x}_i, \tilde{A}) \tag{40}$$

4 Numerical Results

The datasets “Athleticism at Oviedo University” [12], “Diagnosis of Dyslexia” [11], “Ice adhesion strength” [1], “Car” [2], and “Barcelona’s water distribution” [6,15] are used to compare the proposed method against the results of other approaches. The main characteristics of these datasets are summarized in Table 1, where “Ex” represents the number of examples, “Att.” is the number of attributes, “Classes” is the number of classes, and “%Classes” is the fraction of patterns of each class. All these datasets are available in the repository <https://ccia35.edv.uniovi.es/datasets>.

All experiments were repeated 100 times from bootstrap resamples of the training set. The test set comprises “out of the bag” elements. Each test partition is repeated 1000 times for different random crisp selections. The following algorithms, along with their learning parameters, were used:

- GCCL-LQD [11] and Boosting-LQD [13]: population size 100, crossover probability 0.9, mutation probability 0.1, 200 generations, 5 labels/variable, uniform fuzzy partitions.
- FURIA-LQD: Without learning parameters.

In Table 2 it can be shown how the improvement over previous approaches for the most complex problems (Dyslexic-12, Car) is remarkable. The performance of the new classifier improves that of Boosting in most of the high-dimensional problems (Ice-shedding, Dyslexic-12, Car, Water), however FURIA does not seem to improve the accuracy of Boosting for Ice-7, Ice-8 and the low-dimensional problem “Athleticism”.

Table 1. Summary descriptions of the LQD datasets

Dataset	Ex.	Atts.	Classes	%Classes
B200mlI [12]	19	4	2	((0.47,0.73],[0.26,0.52])
B200mlP [12]	19	5	2	((0.47,0.73],[0.26,0.52])
Long [12]	25	4	2	([36,64],[36,64])
BLong [12]	25	4	2	([36,64],[36,64])
100mlI [12]	52	4	2	((0.44,0.63],[0.36,0.55])
100mlP [12]	52	4	2	((0.44,0.63],[0.36,0.55])
B100mlI [12]	52	4	2	((0.44,0.63],[0.36,0.55])
B100mlP [12]	52	4	2	((0.44,0.63],[0.36,0.55])
Ice7-6 [1]	42	7	3	((0.47,0.54],[0.19,0.30],[0.21,0.26])
Ice7-4 [1]				
Ice8-7 [1]	42	8	3	((0.47,0.54],[0.19,0.30],[0.21,0.26])
Ice8-5 [1]				
Ice-shedding [1]	42	7	2	((0.47,0.54],[0.46,0.53])
Dyslexic-12 [11]	65	12	4	((0.32,0.43],[0.07,0.16],[0.24,0.35],[0.12,0.35])
Car [2]	33	8	4	(0.30,0.242,0.242,0.212)
Water_4 [6,15]	316	4	2	(0.705,0.294)

Table 2. Behaviour of GCCL-LQD [11], Boost-LQD [13] and the new proposal (FURIA-LQD) in several datasets

	GCCL-LQD	Boost-LQD	FURIA-LQD
	$AccT_{st}$	$AccT_{st}$	$AccT_{st}$
Atheletics at Oviedo University			
100mlP	[0.640,0.824]	[0.642,0.820]	[0.601,0.780]
100mlI	[0.622,0.824]	[0.624,0.830]	[0.592,0.793]
B100mlP	[0.651,0.840]	[0.650,0.839]	[0.609,0.798]
B100mlI	[0.631,0.828]	[0.644,0.842]	[0.568,0.766]
B200mlP	[0.520,0.738]	[0.594,0.812]	[0.590,0.808]
B200mlI	[0.527,0.768]	[0.585,0.829]	[0.564,0.809]
Long	[0.410,0.679]	[0.492,0.760]	[0.528,0.746]
BLong	[0.375,0.674]	[0.470,0.770]	[0.464,0.764]
Ice adhesion strength			
Ice7-6	[0.512,0.596]	[0.548,0.631]	[0.526,0.606]
Ice7-4	[0.522,0.597]	[0.572,0.651]	[0.528,0.602]
Ice8-7	[0.525,0.597]	[0.554,0.628]	[0.545,0.621]
Ice8-5	[0.507,0.566]	[0.550,0.618]	[0.534,0.608]
Ice-shedding	[0.550,0.619]	[0.639,0.708]	[0.713,0.782]
Diagnostic of dyslexic			
Dyslexic-12	[0.335,0.475]	[0.335,0.464]	[0.434,0.600]
Car			
Car	[0.389,0.389]	[0.436,0.436]	[0.608,0.608]
Barcelona's water distribution			
Water_4	[0.713,0.713]	[0.602,0.602]	[0.651,0.651]
Mean	[0.527,0.670]	[0.559,0.702]	[0.566,0.709]

All in all, later in this section it will be shown that these differences are not statistically relevant, thus it can assumed that the accuracies of FURIA and Adaboost are roughly the same. However,

- FURIA is much faster than Boosting. The combined learning + validation time of FURIA was about 12 times faster than Adaboost or GCCL for the datasets mentioned in this section (5 minutes vs. 1 hour). FURIA is the alternative of choice when computational resources are limited.

- The number of rules of the classifiers produced by FURIA are also much lower. The highest number of rules produced by this algorithm for the studied datasets was of 15, while boosting and GCCL obtain knowledge bases comprising hundreds of rules for the largest problems. That is to say, the linguistic quality of the results of FURIA is much better. It is also remarked that the number of labels for each variable must be determined by trial and error in Boosting and GCCL, but FURIA determines this parameter automatically.

4.1 Statistical Assessment of the Results

The differences in linguistic quality need not to be studied with statistical tests because FURIA-generated knowledge bases were uniformly smaller for all executions of the algorithm. The statistical relevance of the differences in accuracy is assessed with bootstrap tests for LQD, following the experimental design proposed in [10]. The null hypothesis of this test is that the average number of misclassifications for each dataset does not depend on the algorithm. In Table 3 is shown that the mentioned advantages of FURIA over Boosting for high-dimensional datasets are compensated by the results for low-dimensional datasets thus the differences are not significant for a 95% confidence level.

Table 3. Null hypothesis: The expected error is the same than the average

Conditions	GCCL-LQD	BOOST-LQD	FURIA-LQD
$[q-a, q+a]$	[0.527,0.670]	[0.559,0.702]	[0.566,0.709]
$F^*_{(q-a)} \in$	[0,0.025]	[0,0.83]	[0,0.93]
$F^*_{(q+a)} \in$	[0.025,1]	[0.85,1]	[0.94,1]
	INCONCLUSIVE	INCONCLUSIVE	INCONCLUSIVE

5 Concluding Remarks and Future Work

The definition of the algorithm “FURIA” for learning fuzzy rule-based classifiers has been extended to LQD in this contribution. First results seem to show that this algorithm is preferred over boosting or GCCL when computing resources are limited. The linguistic quality of the outcome is also better. However, the accuracy of Boosting can still be higher for some datasets.

In future works, further comparisons should be made that involve the learning time. It is expected that FURIA improves over the alternatives in scenarios with a limited time for evolving a knowledge base, and the results obtained so far seem to confirm this. Lastly, the linguistic quality has been studied on the basis that a small number of rules is better, however the scattered fuzzy partitions produced by FURIA might not be regarded as “human understandable” by most metrics of linguistic quality, that could be included in the analysis.

References

1. Brouwers, E., Peterson, A., Palacios, J.L., Centolanza, L.: Ice Adhesion Strength Measurements for Rotor Blade Edge Materials. In: 67th Annual Forum Proceedings - American Helicopter Society, Virginia Beach, VA (2011)
2. De Carvalho, F.A.T., Souza, R.M., Chavent, M., Lechevallier, Y.: Adaptive Hausdorff distances and dynamic clustering of symbolic interval data. *Pattern Recognition* 27, 167–179 (2006)
3. Cohen, W.: Fast effective rule induction. In: Frieditis, A., Russel, S. (eds.) *Proceeding of the 12th International Conference on Machine Learning, ICML*, pp. 115–123 (1995)
4. Couso, I., Sanchez, L.: Higher order models for fuzzy random variables. *Fuzzy Sets and Systems* 159, 237–258 (2008)
5. Demsar, J.: Statistical comparisons of classifiers over multiple datasets. *Journal of Machine Learning Research* 7, 1–30 (2006)
6. Hedjazi, L., Aguilar-Martin, J., Le Lann, M.V.: Similarity-margin based feature selection for symbolic interval data. *Pattern Recognition Letters* 32, 578–585 (2011)
7. Hühn, J.C., Hüllermeier, E.: FURIA: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery* 19, 293–319 (2009)
8. Hühn, J.C., Hüllermeier, E.: FURIA: Fuzzy Unordered Rule Induction Algorithm (2009), <http://www.uni-marburg.de/fb12/kebi/research/software/furia>
9. Hühn, J.C., Hüllermeier, E.: An analysis of the FURIA algorithm for fuzzy rule induction. In: Koronacki, J., Raś, Z.W., Wierzchoń, S.T., Kacprzyk, J. (eds.) *Advances in Machine Learning I. SCI*, vol. 262, pp. 321–344. Springer, Heidelberg (2010)
10. Otero, J., Sánchez, L., Couso, I., Palacios, A.: Bootstrap analysis of multiple repetitions of experiments using an interval value multiple comparison procedure. *Journal of Computer and System Sciences* (accepted), doi:10.1016/j.jcss.2013.03.009
11. Palacios, A., Sánchez, L., Couso, I.: Diagnosis of dyslexia with low quality data with genetic fuzzy systems. *International Journal on Approximate Reasoning* 51, 993–1009 (2010)
12. Palacios, A., Sánchez, L., Couso, I.: Future performance modelling in athleticism with low quality data-based GFSSs. *Journal of Multivalued Logic and Soft Computing* 17(2-3), 207–228 (2011)
13. Palacios, A., Sánchez, L., Couso, I.: Boosting of fuzzy rules with low quality data. *Journal of Multiple-Valued Logic and Soft Computing* 19(5-6), 591–619 (2012)
14. Sánchez, L., Couso, I., Casillas, J.: Genetic learning of fuzzy rules on low quality data. *Fuzzy Sets and Systems* 160(17), 2524–2552 (2009)
15. Quevedo, J., Puig, V., Cembrano, G., Blanch, J., Aguilar, J., Saporta, D., Benito, G., Hedo, M., Molina, A.: Validation and reconstruction of flow meter data in the Barcelona water distribution network. *J. Control Eng. Practice* 18, 640–651 (2010)
16. Teich, J.: Pareto-front exploration with uncertain objectives. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) *EMO 2001. LNCS*, vol. 1993, pp. 314–328. Springer, Heidelberg (2001)

Author Index

- Abdullah, Rosni 345
Achuthan, Anusha 92
Alberola, Juan M. 161
Alkan, Ozgur Kirmemis 181
Amanatiadis, Angelos 324
Amandi, Analía 376
Analide, Cesar 252
Andreica, Anca 669
Apolloni, Bruno 540
Aranda-Corral, Gonzalo A. 202
Ayerdi, Borja 491
- Banković, Zorana 401
Barbosa, Ernesto 71
Barreto, Guilherme A. 588
Barros, Ana Luiza B.P. 588
Baruque, Bruno 334
Berlanga, Antonio 140
Bernardos, Ana M. 242
Borrego-Díaz, Joaquín 202
Botía, Juan A. 41
Botón-Fernández, María 366
Botti, Vicente 21
Burduk, Robert 132
- Caamaño, Pilar 390
Caha, Jan 548
Calvo, Guillermo 334
Campillo-Sanchez, Pablo 41
Cárdenas-Montes, Miguel 356
Carneiro, Davide 222
Casar, José R. 242
Castejon, Pablo 530
Castriello, Francisco Prieto 366
Charte, Francisco 150
Chira, Camelia 659, 669
Cho, Sung-Bae 618
Chyzyk, Darya 482
Cilla, Rodrigo 140
Corchado, Emilio 280, 334
Corrigan, Derek 112
Couso, Ines 679
Curcin, Vasa 112
- de Carvalho, André C.P.L.F. 629
de la Hoz, Eduardo 103
de la Hoz, Emiro 103
Delaney, Brendan 112
del Jesus, María José 150
del Val, Elena 21, 161
De Pietro, Giuseppe 81, 269, 558
Díaz, Irene 232
Díaz, Julia 51
Dorronsoro, José R. 51
Duro, Richard J. 390
Dvorský, Jiří 548
- Esgin, Eren 191
Esposito, Massimo 81, 269, 558
Etxeberria-Agiriano, Ismael 510
- Fernández, Ángela 51
Fernández-de-Alba, José M. 31
Filasiak, Robert 212
Fiosina, Jelena 639
Fiosins, Maksims 639
Frank, Jeremy 530
Fuentes-Fernández, Rubén 31
Fuksz, Levente 649
- Gala, Yvonne 51
Galán-Páez, Juan 202
Galar, Mikel 568
García, Luís Paulo F. 629
García-Magariño, Iván 11
Gasteratos, Antonios 324
Gebhardt, Jörg 598
Gómez-Sanz, Jorge J. 11, 41
González, Silvia 659
Graña, Manuel 482, 491, 540
Guevara, Elizabeth 304
Gutiérrez, Pedro Antonio 472, 500
Guzmán-Alvarez, César 451, 530
- Hajdu, Andras 314
Hajdu, Lajos 314
Herrera, Francisco 150, 568
Herrero, Álvaro 280

- Hervás-Martínez, Cesar 472, 500
 Hetmańczyk, Mariusz Piotr 262
- Iannaccone, Marco 81
 Iglesias, Josué 242
- Julian, Vicente 161
- Kajdanowicz, Tomasz 112, 431
 Karagoz, Pinar 171, 181, 191
 Kavurucu, Yusuf 171
 Kazienko, Przemyslaw 112
 Kepski, Michal 294
 Kim, Yong-Joong 618
 Kostavelis, Ioannis 324
 Kovacs, Laszlo 314
 Krawczyk, Bartosz 462
 Kruse, Rudolf 598
 Kwolek, Bogdan 294
- López, Vivian F. 122
 López-García, Pedro 401
 Lopez-Guede, Jose Manuel 510
 Lorena, Ana C. 629
 Louçã, Jorge 411
 Luckner, Marcin 212
 Luengo, Julián 568
- Maiora, Josu 491
 Majak, Marcin 421
 Manzanedo, Miguel A. 280
 Martínez-Álvarez, Francisco 578
 Martínez-Ballesteros, María 578
 Martyna, Jerzy 441
 Mercaderes, Roxana Danger 112
 Michalski, Piotr 262
 Minutolo, Aniello 269
 Mohamed Yusoff, Syarifah Adilah 345
 Molina, José M. 140
 Mollá, Mercedes 356
 Moreno, Maily 1
 Moreno, María N. 122
 Muñoz, María Dolores 122
 Mutlu, Alev 171
- Nachyla, Beata 608
 Neves, João 71
 Neves, José 222
 Novais, Paulo 71, 222, 252
 Nuñez-Gonzalez, J. David 540
- Oliveira, Tiago 71
 Onaindia, Eva 530
 Oniga, Stefan 520
 Ortega, Julio 103
 Ortiz, Andrés 103
- Palacios, Ana Maria 679
 Patricio, Miguel A. 140
 Pavón, Juan 1, 31
 Pérez, Arturo 280
 Pérez-Ortiz, María 472
 Pimenta, André 222
 Pop, Petrica C. 649
 Popiel, Adrian 112
 Pop-Sitar, Petrica 520
 Pota, Marco 558
- Rajeswari, Mandava 92
 Ramos, Vitorino 411
 Rebollo, Miguel 21
 Reinbacher, Thomas 451
 Riquelme, José C. 578
 Rivera, Antonio 150
 Rodrigues, David M.S. 411
 Rodríguez-Muniz, Luis J. 232
 Rosete, Alejandro 1
- Sáez, José A. 568
 Sáiz, Lourdes 280
 Sánchez, Angel Luis 122
 Sanchez, Luciano 679
 Sanchez-Anguix, Victor 161
 Sánchez-Monedero, Javier 500
 Schmidt, Fabian 598
 Sedano, Javier 659
 Segrera, Saddys 122
 Silva, Fábio 252
 Simić, Dragan 61
 Simić, Svetlana 61
 Soler, Jean Karl 112
 Sossa, Humberto 304
 Suberbiola, Aaron 510
 Svirčević, Vasa 61
 Świder, Jerzy 262
 Szymański, Piotr 431
- Toman, Henrietta 314
 Trajdos, Pawel 132
 Trejo, José M. 659

- Troiano, Luigi 232
Troncoso, Alicia 578

Van Caesbroeck, Bren 510
Varela, Gervasio 390
Vega-Rodríguez, Miguel Á. 356, 366
Venkat, Ibrahim 345
Villar, José Ramón 659

Wendler, Jan 598
Woźniak, Michał 462

Yannibelli, Virginia 376

Zolnierek, Andrzej 421
Zulueta, Ekaitz 510