

Jose A. Lozano
José A. Gámez
José A. Moreno (Eds.)

LNAI 7023

Advances in Artificial Intelligence

14th Conference of the Spanish Association
for Artificial Intelligence, CAEPIA 2011
La Laguna, Spain, November 2011, Proceedings

 Springer

Lecture Notes in Artificial Intelligence 7023

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

Jose A. Lozano José A. Gámez
José A. Moreno (Eds.)

Advances in Artificial Intelligence

14th Conference of the Spanish Association
for Artificial Intelligence, CAEPIA 2011
La Laguna, Spain, November 7-11, 2011
Proceedings

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Jose A. Lozano
University of the Basque Country
Computer Science School
P^o Manuel de Lardizabal 1
20018 Donostia-San Sebastian, Spain
E-mail: ja.lozano@ehu.es

José A. Gámez
University of Castilla-La Mancha
Computing Systems Department
Campus Universitario s/n
02071 Albacete, Spain
E-mail: jose.gamez@uclm.es

José A. Moreno
University of La Laguna
Department of Statistics, O.R. and Computation
38271 La Laguna, S.C. Tenerife, Spain
E-mail: jamoreno@ull.es

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-25273-0 e-ISBN 978-3-642-25274-7
DOI 10.1007/978-3-642-25274-7
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011940439

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, 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.

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 contains a selection of the papers accepted for oral presentation at the 14th Conference of the Spanish Association for Artificial Intelligence (CAEPIA 2011), held in La Laguna (Canary Islands, Spain), November 7–11, 2011. This was the 14th biennial conference in the CAEPIA series, which was started back in 1985. Previous editions took place in Madrid, Alicante, Málaga, Murcia, Gijón, Donostia, Santiago de Compostela, Salamanca and Seville.

CAEPIA 2011 included all the scientific events and activities of big conferences. The scientific program consisted of technical presentations of accepted papers. We also had three renowned keynote speakers, J.L. Verdegay (University of Granada, Spain), T. Mitchell (Carnegie Mellon University, USA) and M.A. Vila Miranda (University of Granada, Spain), who gave impressive plenary talks. The conference also featured seven workshops on different hot topics of the field. Three two-hour tutorials were offered in the conference to introduce the audience to advanced topics in artificial intelligence. Finally, we would like to highlight the doctoral consortium, an informal forum where the PhD students present their work and senior researchers give them feedback.

With the permanent goal of making CAEPIA a high-quality conference, and following the model of current demanding AI conferences, we organized the review process for CAEPIA papers in the following way. The Scientific Committee was structured in two levels. At the first level we distributed the AI knowledge in 10 areas and named a Track Chair for each one. These Track Chairs are well-known members of the AI community affiliated to Spanish and non-Spanish universities and research centers. Secondly, there was a Program Committee consisting of almost 100 members (30 non-Spanish institutions). Each paper was assigned to three Program Committee members, who made the reviews (following the double-blind model), and to the Track Chair, who supervised these reviews. On the basis of the reviews, the Track Chairs took the final decision about the papers.

We received 149 submissions. After the review process, only 50 papers were accepted and selected to be published in this volume. We would like to acknowledge the work done by the Scientific Committee members in the review and discussion of the submissions, and by the authors to improve the quality of AI research. We would also like to thank the invited speakers and the professors in charge of the tutorials for their participation in the conference. Last but not least, we would like to thank the Organizing Committee members for their hard work, the University of La Laguna, our sponsors and AEPIA for their support.

August 2011

Jose A. Lozano
José A. Gámez
José A. Moreno

Organization

Executive Committee

Jose A. Lozano	University of the Basque Country
José A. Gámez	University of Castilla-La Mancha
José A. Moreno	University of La Laguna

Senior Program Committee

Alejandro Bellogín, Spain	Jorge Civera, Spain
Alicia Troncoso, Spain	José M. Peña, Sweden
Asunción Gómez-Pérez, Spain	José Marcos Moreno, Spain
Boris Villazón-Terrazas, Spain	José M. Juarez, Spain
Domingo Savio Rodríguez-Baena, Spain	José C. Riquelme, Spain
Enrique Herrera-Viedma, Spain	José Palma, Spain
Filiberto Pla, Spain	Manuel Ojeda-Aciego, Spain
Grzegorz J. Nalepa, Poland	María Guijarro, Spain
Hector Geffner, Spain	Nicolás García-Pedrajas, Spain
Héctor Pomares, Spain	Norberto Díaz-Díaz, Spain
Iván Cantador, Spain	Oscar Corcho, Spain
Joachim Baumeister, Germany	Oscar Cordón, Spain
Joaquín Cañadas, Spain	Pablo Castells, Spain
	Sascha Ossowski, Spain

Program Committee

Agustín Valverde, Spain	Carlos Cotta, Spain
Alberto Bugarín, Spain	Carlos Damásio, Portugal
Alberto Fernández, Spain	Carlos Linares, Spain
Alexander Mendiburu, Spain	Carme Torras, Spain
Alexandre Aussem, France	Carmen Paz Suárez, Spain
Alfons Juan, Spain	César García-Osorio, Spain
Andrés Cano, Spain	César Hervás, Spain
Ángel Sappa, Spain	Chris Cornelis, Belgium
Antonio Fernández, Spain	Christiam Blum, Spain
Antonio Salmerón, Spain	Christian Guttman, Australia
Basilio Sierra, Spain	Colin Fyfe, UK
Beatriz López, Spain	Concha Bielza, Spain
Belén Melián, Spain	Daniel Borrajo, Spain
Carlos Coello, Mexico	David Pearce, Spain

David Pelta, Spain
Domingo Ortiz, Spain
Emilio Corchado, Spain
Enric Cervera, Spain
Enrique Alba, Spain
Francesc J. Ferri, Spain
Francesc Moreno, Spain
Francesco Buccafurri, Italy
Francisco Casacuberta, Spain
Francisco Herrera, Spain
Francisco Mario Hernández, Spain
Frank Hoffmann, Germany
Gonzalo Cerruela, Spain
Hermann Ney, Germany
Humberto Bustince, Spain
Ines Lynce, Portugal
Iñaki Inza, Spain
Jaume Bacardit, UK
Jesús Andrés-Ferrer, Spain
Jesús Medina, Spain
Joan Serrat, Spain
Joao Gama, Portugal
Jordi Vitria, Spain
José Angel Olivás, Spain
Jose B. Mariño, Spain
José Jesús Guerrero, Spain
José L. Balcázar, Spain
José Luis Verdegay, Spain
Jose M. Gutierrez, Spain
José Manuel Cadenas, Spain
José María Martínez, Spain
Jose M. Puerta, Spain
José Miguel Sanchiz, Spain
José R. Dorronsoro, Spain
Juan J. Rodríguez, Spain
Juan José del Coz, Spain
Juan Manuel Corchado, Spain
Juan Manuel Fernández, Spain
Juan Manuel Molina, Spain
Juan Pavón, Spain
Kishan Mehrotra, USA
L. Enrique Sucar, Mexico
Lawrence Mandow, Spain
Lluís Godó, Spain
Luciano Sánchez, Spain
Luis Castillo, Spain
Luis de la Ossa, Spain
Luis M. de Campos, Spain
Luis Magdalena, Spain
Marc Esteve, Spain
Marcello Federico, Italy
María José del Jesus, Spain
María Teresa Lamata, Spain
Michael Fink, Austria
Mikel Forcada, Spain
Miquel Sanchez, Spain
Pablo Castells, Spain
Pablo Cordero, Spain
Pablo Varona, Spain
Pedro Cabalar, Spain
Pedro Larrañaga, Spain
Pedro Meseguer, Spain
Philipp Koehn, UK
Ramiro Varela, Spain
Rasa Jurgelenaite, The Netherlands
Raúl Giráldez, Spain
Ricard Gavaldà, Spain
Richard Duro, Spain
Robert Castelo, Spain
Roberto Ruiz, Spain
Roberto Santana, Spain
Roque Marín, Spain
Sancho Salcedo, Spain
Sebastián Sardina, Australia
Sebastián Ventura, Spain
Serafín Moral, Spain
Slawomir Zadrozny, Poland
Thomas Stützle, Belgium
Ulrich Bodenhofer, Austria
Umberto Straccia, Italy
Vicenç Torra, Spain
Vicent Botti, Spain
Vicente Julian, Spain
Vicente Matellán, Spain
Vitaly Schetinin, UK
Wesam Barbakh, Israel

Organizing Committee

Airam Expósito	Dionisio Pérez	Jesús David Beltrán
Belén Melián	Eduardo Lalla	Jezabel Molina
Candelaria Hernández	Elena Sánchez	José Luis González
Cándido Caballero	F. Javier Martínez	Julio Brito
Carlos Echegoyen	José Marcos Moreno	Patricio García
Christopher Expósito	Javier Rodríguez	Pino Caballero

Sponsors

Gobierno de España-Ministerio de Ciencia e Innovación

Gobierno de Canarias-Agencia Canaria de Investigación, Innovación y Sociedad de la Información

Cabildo de Tenerife

Ayuntamiento de San Cristobal de La Laguna

Universidad de La Laguna

Table of Contents

Agent-Based and Multiagent Systems

Evaluating a Reinforcement Learning Algorithm with a General Intelligence Test	1
<i>Javier Insa-Cabrera, David L. Dowe, and José Hernández-Orallo</i>	
Evaluation of an Automated Mechanism for Generating New Regulations	12
<i>Javier Morales, Maite López-Sánchez, and Marc Esteva</i>	
A Multi-agent System for Incident Management Solutions on IT Infrastructures	22
<i>Elena Sánchez-Nielsen, Antonio Padrón-Ferrer, and Francisco Marreo-Estévez</i>	
Market Self-organization under Limited Information	32
<i>Gregor Reich</i>	
Solving Sequential Mixed Auctions with Integer Programming	42
<i>Boris Mikhaylov, Jesus Cerquides, and Juan A. Rodriguez-Aguilar</i>	

Machine Learning

Global Feature Subset Selection on High-Dimensional Datasets Using Re-ranking-based EDAs	54
<i>Pablo Bermejo, Luis de La Ossa, and Jose M. Puerta</i>	
A Comparison of Two Strategies for Scaling Up Instance Selection in Huge Datasets	64
<i>Aida de Haro-García, Javier Pérez-Rodríguez, and Nicolás García-Pedrajas</i>	
C4.5 Consolidation Process: An Alternative to Intelligent Oversampling Methods in Class Imbalance Problems	74
<i>Iñaki Albisua, Olatz Arbelaiz, Ibai Gurrutxaga, Javier Muguerza, and Jesús M. Pérez</i>	
Scalability Analysis of ANN Training Algorithms with Feature Selection	84
<i>Verónica Bolón-Canedo, Diego Peteiro-Barral, Amparo Alonso-Betanzos, Bertha Guijarro-Berdiñas, and Noelia Sánchez-Marroño</i>	

Using Model Trees and Their Ensembles for Imbalanced Data	94
<i>Juan J. Rodríguez, José F. Díez-Pastor, César García-Osorio, and Pedro Santos</i>	
Instance Selection for Class Imbalanced Problems by Means of Selecting Instances More Than Once	104
<i>Javier Pérez-Rodríguez, Aida de Haro-García, and Nicolás García-Pedrajas</i>	
On the Effectiveness of Distributed Learning on Different Class-Probability Distributions of Data	114
<i>Diego Peteiro-Barral, Bertha Guijarro-Berdiñas, and Beatriz Pérez-Sánchez</i>	
On the Learning of ESN Linear Readouts	124
<i>Carlos M. Alaíz and José R. Dorronsoro</i>	
Learning Naive Bayes Models for Multiple-Instance Learning with Label Proportions	134
<i>Jerónimo Hernández and Iñaki Inza</i>	
The von Mises Naive Bayes Classifier for Angular Data	145
<i>Pedro L. López-Cruz, Concha Bielza, and Pedro Larrañaga</i>	
Unravelling the Yeast Cell Cycle Using the TriGen Algorithm	155
<i>David Gutiérrez-Avilés, Cristina Rubio-Escudero, and José C. Riquelme</i>	
Pattern Recognition in Biological Time Series	164
<i>Francisco Gómez-Vela, Francisco Martínez-Álvarez, Carlos D. Barranco, Norberto Díaz-Díaz, Domingo Savio Rodríguez-Baena, and Jesús S. Aguilar-Ruiz</i>	
Knowledge Representation, Logic, Search and Planning	
On the Expressive Power of First Order-Logic Extended with Allen's Relations in the Strict Case	173
<i>Willem Conradie and Guido Sciavicco</i>	
Using the Relaxed Plan Heuristic to Select Goals in Oversubscription Planning Problems	183
<i>Angel García-Olaya, Tomás de la Rosa, and Daniel Borrajo</i>	
Optimally Scheduling a Job-Shop with Operators and Total Flow Time Minimization	193
<i>María R. Sierra, Carlos Mencía, and Ramiro Varela</i>	

OntoMetaWorkflow: An Ontology for Representing Data and Users in Workflows	203
<i>Alvaro E. Prieto, Adolfo Lozano-Tello, and José Luis Redondo-García</i>	

Architecture for the Use of Synergies between Knowledge Engineering and Requirements Engineering	213
<i>José del Sagrado, Isabel M. del Águila, and Francisco J. Orellana</i>	

Multidisciplinary Topics and Applications

Dynamic Bayesian Network Factors from Possible Conflicts for Continuous System Diagnosis	223
<i>Carlos J. Alonso-Gonzalez, Noemi Moya, and Gautam Biswas</i>	

Planning and Execution in a Personalised E-Learning Setting	233
<i>Lluvia Morales, Antonio Garrido, and Ivan Serina</i>	

Heuristic Multiobjective Search for Hazmat Transportation Problems ...	243
<i>Enrique Machuca, Lawrence Mandow, José Luis Pérez de la Cruz, and Antonio Iovanella</i>	

Topography of Functional Connectivity in Human Multichannel Electroencephalogram during Second Language Processing	253
<i>Ernesto Pereda, Susanne Reiterer, and Joydeep Bhattacharya</i>	

A Summary on the Study of the Medium-Term Forecasting of the Extra-Virgen Olive Oil Price	263
<i>Antonio Jesús Rivera, María Dolores Pérez-Godoy, María José del Jesus, Pedro Pérez-Recuerda, María Pilar Frías, and Manuel Parras</i>	

SMS Normalization: Combining Phonetics, Morphology and Semantics	273
<i>Jesús Oliva, José Ignacio Serrano, María Dolores del Castillo, and Ángel Iglesias</i>	

Vision and Robotics

Multiscale Extension of the Gravitational Approach to Edge Detection	283
<i>Carlos Lopez-Molina, Bernard De Baets, Humberto Bustince, Edurne Barrenechea, and Mikel Galar</i>	

A Study of the Suitability of Evolutionary Computation in 3D Modeling of Forensic Remains	293
<i>José Santamaría, Oscar Cordón, Sergio Damas, Jose M. García-Torres, and Fernando Navarro</i>	

L-System-Driven Self-assembly for Swarm Robotics 303
Fidel Aznar, Mar Pujol, and Ramón Rizo

An Study on Ear Detection and Its Applications to Face Detection 313
Modesto Castrillón-Santana, Javier Lorenzo-Navarro, and Daniel Hernández-Sosa

A Combined Strategy Using FMCDM for Textures Segmentation in Hemispherical Images from Forest Environments 323
P. Javier Herrera, Gonzalo Pajares, and María Guijarro

Getting NDVI Spectral Bands from a Single Standard RGB Digital Camera: A Methodological Approach 333
Gilles Rabatel, Nathalie Gorretta, and Sylvain Labbé

Soft Computing

Combining Neighbourhoods in Fuzzy Job Shop Problems 343
Jorge Puente, Camino R. Vela, and Inés González-Rodríguez

Learning Cooperative TSK-0 Fuzzy Rules Using Fast Local Search Algorithms 353
Javier Cózar, Luis de la Ossa, and Jose M. Puerta

Weighted Tardiness Minimization in Job Shops with Setup Times by Hybrid Genetic Algorithm 363
Miguel A. González, Camino R. Vela, and Ramiro Varela

Interval-Valued Fuzzy Sets for Color Image Super-Resolution 373
Aranzazu Jurio, José Antonio Sanz, Daniel Paternain, Javier Fernandez, and Humberto Bustince

An Evolutionary Multiobjective Constrained Optimisation Approach for Case Selection: Evaluation in a Medical Problem 383
Eduardo Lupiani, Fernando Jimenez, José M. Juarez, and José Palma

Web Intelligence and Information Retrieval

The VoiceApp System: Speech Technologies to Access the Semantic Web 393
David Griol, José Manuel Molina, and Víctor Corrales

A Cluster Based Pseudo Feedback Technique which Exploits Good and Bad Clusters 403
Javier Parapar and Álvaro Barreiro

SAHN with SEP/COP and SPADE, to Build a General Web Navigation Adaptation System Using Server Log Information	413
<i>Olatz Arbelaitz, Ibai Gurrutxaga, Aizea Lojo, Javier Muguerza, and Iñigo Perona</i>	
A New Criteria for Selecting Neighborhood in Memory-Based Recommender Systems	423
<i>Sergio Cleger-Tamayo, Juan M. Fernández-Luna, and Juan F. Huete</i>	
Extended Precision Quality Measure for Recommender Systems	433
<i>Fernando Ortega, Antonio Hernando, and Jesús Bobadilla</i>	
A Tool for Link-Based Web Page Classification	443
<i>Inma Hernández, Carlos R. Rivero, David Ruiz, and Rafael Corchuelo</i>	
Information Extraction for Standardization of Tourism Products	453
<i>Nuno Miranda, Ricardo Raminhos, Pedro Seabra, Teresa Gonçalves, José Saias, and Paulo Quaresma</i>	
Improving Collaborative Filtering in Social Tagging Systems	463
<i>Felice Ferrara and Carlo Tasso</i>	
Ontology-Driven Method for Integrating Biomedical Repositories	473
<i>José Antonio Miñarro-Giménez and Jesualdo Tomás Fernández-Breis</i>	
Lightweighting the Web of Data through Compact RDF/HDT	483
<i>Javier D. Fernández, Miguel A. Martínez-Prieto, Mario Arias, Claudio Gutierrez, Sandra Álvarez-García, and Nieves R. Brisaboa</i>	
Query Expansion Methods and Performance Evaluation for Reusing Linking Open Data of the European Public Procurement Notices	494
<i>Jose María Álvarez, José Emilio Labra, Ramón Calmeau, Ángel Marín, and José Luis Marín</i>	
Author Index	505

Evaluating a Reinforcement Learning Algorithm with a General Intelligence Test

Javier Insa-Cabrera¹, David L. Dowe², and José Hernández-Orallo¹

¹ DSIC, Universitat Politècnica de València, Spain
{jinsa,jorallo}@dsic.upv.es

² Clayton School of Information Technology, Monash University, Australia
david.dowe@monash.edu

Abstract. In this paper we apply the recent notion of anytime universal intelligence tests to the evaluation of a popular reinforcement learning algorithm, Q-learning. We show that a general approach to intelligence evaluation of AI algorithms is feasible. This top-down (theory-derived) approach is based on a generation of environments under a Solomonoff universal distribution instead of using a pre-defined set of specific tasks, such as mazes, problem repositories, etc. This first application of a general intelligence test to a reinforcement learning algorithm brings us to the issue of task-specific vs. general AI agents. This, in turn, suggests new avenues for AI agent evaluation and AI competitions, and also conveys some further insights about the performance of specific algorithms.

1 Introduction

In order to evaluate progress in AI, intelligence and performance tests are crucial. We know about many AI competitions held in many different realms (learning, planning, robotics, games, ...). Most of them, however, are just constructed as a set of specific tasks. While many of these competitions modify and extend the set of tasks each year in order to cover a broader view of the field and avoid competition-specialisation, the information which is obtained from these competitions is still limited. Winners are frequently the teams which have devoted more time understanding the nuts and bolts of the competition and to (correspondingly) tuning their algorithms for the tasks. Also, the ‘complexity’ of each task is always quantified or estimated in an informal or ad-hoc way, so it is very difficult to compare results across different algorithms and competitions.

An alternative proposal for intelligence and performance evaluation is based on the notion of universal distribution [12] and the related algorithmic information theory (a.k.a. Kolmogorov complexity) [10]. Note that any universal distribution does not assign the same probability to all objects (which would be 0, since there are infinitely many), but it gives higher probability to objects with smaller descriptions. Using this theory, we can define a universal distribution of tasks for a given AI realm, and sort them according to their (objective) complexity. Some early works have developed these ideas to construct intelligence tests.

First, [1] suggested the introduction of inductive inference problems in a somehow *induction-enhanced* or *compression-enhanced* Turing Test [15]. Second, [3] derived intelligence tests (C-tests) as sets of sequence prediction problems which were generated by a universal distribution, and the result (the intelligence of the agent) was a sum of performances for a range of problems of increasing complexity. The complexity of each sequence was derived from its Kolmogorov complexity (a Levin variant was used). This kind of problem (discrete sequence prediction), although typical in IQ tests, is a narrow AI realm. In fact, [11] showed that relatively simple algorithms could score well at IQ tests (and, as a consequence, at C-tests). In [3] the suggestion of using interactive tasks where “rewards and penalties could be used instead” was made. Later, Legg and Hutter (e.g. [7, 8]) gave a precise definition to the term “Universal Intelligence”, as a sum (or weighted average) of performances in all the possible environments. Environments are understood as is custom in reinforcement learning. However, in order to make the extension from (static) sequences to (dynamic) environments, several issues had to be solved first. In [6], the problem of finding a finite sample of environments and sessions is addressed, as well as approximations to Kolmogorov complexity, the inclusion of time, and the proper aggregation of rewards. The theory, however, has not been applied in the form of a real test, to evaluate artificial and biological agents. This is the goal of our paper.

Since these recent approaches are constructed over a reinforcement learning (RL) setting, it seems natural to start evaluating RL algorithms. In fact, RL [14] [20] is a proper and general setting to define and analyse learning agents which interact with an environment through the use of observations, actions and rewards. Hence, RL is not strictly limited to AI agents; non-human animals and humans can be understood in this setting, most especially in the context of evaluation. When trying to pick up a ‘representative’ algorithm to start with, we face a difficult issue, since there is a vast amount of literature on RL algorithms. According to [20], the three most influential algorithms are Temporal Difference (TD) Learning, adaptive Actor-Critics and Q-learning [17]. Here we choose Q-learning and we evaluate it in terms of the theory given in [6] and an environment class defined in [4]. We present here a first implementation of the tests and we evaluate Q-learning using these tests. The use of a general intelligence test for Q-learning provides some interesting insights into how RL algorithms could be evaluated (with a general intelligence test) and also into the viability of the test as a general intelligence test for AI.

The paper is organised as follows. Section 2 briefly describes the theory presented in [6] and the environment class introduced in [4]. Section 3 gives some details on the implementation of the tests, and introduces the types of agents we will evaluate with the test. The next sections perform an experimental evaluation, using a simple example first (section 4), showing the basic experimental results and their relation to complexity (section 5). Section 6 follows with a discussion of the results and related work, and section 7 closes the paper.

2 An Environment Class for a Universal Intelligence Test

Effective testing and evaluation of an individual's ability requires an accurate choice of items in such a way that the tests are discriminative and quantify the capability to be measured. Measuring (machine) intelligence is not different. [6] presents the first general and feasible setting to construct an intelligence test which claims to be valid for both artificial intelligent systems and biological systems, of any intelligence degree and of any speed. The test is not anthropomorphic, is gradual, is anytime and is exclusively based on computational notions, such as Kolmogorov complexity. And it is also meaningful, since it averages the capability of succeeding in different environments. The notion of environment is similar to the general notion which is used in reinforcement learning - by using actions, rewards and observations. The key idea is to order all the possible environments by their Kolmogorov complexity and use this ordering to make samples and construct adaptive tests that can be used to evaluate the intelligence of any kind of agent. The test configures a new paradigm for intelligence measurement which dramatically differs from the current task-oriented and ad-hoc measurement used both in artificial intelligence and psychometrics.

One of the key issues in the previous test is the use of discriminative environments only. That means that environments which may lead to dead-ends, are too slow, or that only allow a few interactions with the agent are ruled out. Additionally, a selection of the remaining environments must be done according to a sample of all the (infinitely many) possible environments. The choice of an unbiased probability distribution to make the sample is then crucial.

As a consequence, the choice of a proper environment class is a crucial issue. The more general the environment class, the better. This is what [4] attempts, a hopefully unbiased environment class (called \mathcal{A}) with spaces and agents with universal descriptive (Turing-complete) power. Basically, the environment class \mathcal{A} considers a space as a graph of cells (nodes) and actions (vertices). Objects and agents can be introduced using Turing-complete languages to generate their movements. The environment class can be summarised as follows:

- Space (Cells and Actions): The space is defined as a directed labelled graph, where each node represents a cell, and arrows represent actions. The topology of the space can vary, since it is defined by a randomly-generated set of rules (using a geometric distribution with $p = 1/2$). The graph is selected to be strongly connected (all cells are reachable from any other cell).
- Agents: Cells can contain agents. Agents can act deterministically (or not) and can be reactive to other agents. Agents perform one action at each interaction of the environment. Every environment must include at least three agents: the evaluated agent, and two special agents *Good* and *Evil*.
- Observations and Actions: Actions allow the evaluated agent (and other agents) to move in the space. Observations show the cell contents.
- Rewards: rewards are generated by means of the two special agents *Good* and *Evil*, which leave rewards in the cells they visit. Rewards are rational numbers in the interval $[-1, 1]$. *Good* and *Evil* have the same pattern for

behaviour except for the sign of the reward (+ for *Good*, – for *Evil*). This makes *Good* and *Evil* symmetric, which ensures that the environment is balanced (random agents score 0 on average) [6]. *Good* and *Evil* are initially placed randomly in different cells (and they cannot share a cell).

For the space (the graph) and also for the behaviour of all the agents, a Turing-complete language based on rewriting rules (Markov algorithms) is proposed.

The environment class \mathcal{A} is shown in [4] to have two relevant properties for a performance test: (1) their environments are always balanced, and (2) their environments are reward-sensitive (there is no sequence of actions such that the agent can be stuck in a heaven or hell situation, where rewards are independent of what the agent may do). As argued in [6], these two properties are very important for the environments to be discriminative and comparable (and hence the results being properly aggregated into a single performance or intelligence score). No other properties are imposed, such as (e.g.) environments being Markov processes or being ergodic.

Several interfaces have been defined so that we can test biological subjects and machines. In this paper we will focus on evaluating machine algorithms. For more details of the environment class \mathcal{A} , see [4].

3 Implementation and Evaluated Agents

Following the definition of the environment class \mathcal{A} , we generate each environment as follows. Spaces are generated by first determining the number of cells n_c , which is given by a number between 2 and 9, using a geometric distribution (i.e. $\text{prob}(n) = 2^{-n}$, and normalising to sum up to 1). Similarly, the number of actions n_a is defined with a geometric distribution between 2 and n_c . Both cells and actions are indexed with natural numbers. There is a special action 0 which connects every cell with itself (i.e., to stay at the cell). The connections between cells are determined using a uniform distribution for each cell, among the possible actions and cells. We consider the possibility that some actions do not lead to any cell. These actions have no effect. A cell which is accessible from another cell (using one action) is called a ‘neighbouring’ or adjacent cell. An example of a randomly generated space can be shown in Fig. 1.

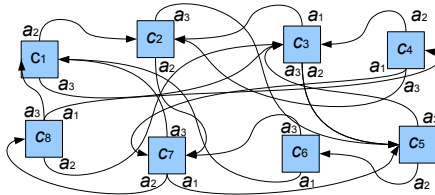


Fig. 1. A space with 8 cells and 4 actions (a_0, a_1, a_2, a_3). Reflexive action a_0 not shown.

The number of cells and actions is, of course, related to the complexity of the space, but not monotonically related to its Kolmogorov complexity (or a computable variant such as Levin’s Kt [9]). Nonetheless, most of the actual grading of environments comes from the behaviour of *Good* and *Evil*. The sequence of actions for *Good* and *Evil* is defined by using a uniform distribution for each element in the sequence, and a geometric distribution ($p = 1/100$) to determine whether to stop the sequence, by using a probability of stopping (p_{stop}). An example of a sequence for the space in Fig. 1 is 203210200, which means the execution of actions a_2, a_0, a_3, a_2 , etc. If *Good* is placed at cell c_5 , the pattern will lead it (via a_2) to c_6 in the next step, since it starts with ‘2’. The agents *Good* and *Evil* take one action from the sequence and execute it for each step in the system. When the actions are exhausted, the sequence is started all over again. If an action is not allowed at a particular cell, the agent does not move.

Given a single environment, evaluation is performed in the following way. Initially, each agent is randomly (using a uniform distribution) placed in a cell. Then, we let *Good*, *Evil*, the evaluated agent and any other agents in the space interact for a certain number of steps. We call this a session. For a session we average the rewards, so giving a score of the agent in the environment.

Although [4] suggests a partially-observable interface, here we will make it fully-observable, i.e., the agents will be able to see all the cells, actions and contents. Rewards are not part of the observation and hence are not shown.

And now we present the agents we will evaluate.

- Random: a random agent is just an agent which chooses randomly among the available actions using a uniform distribution.
- Trivial Follower: this is an agent which looks at the neighbouring cells to see whether *Good* is in one of them. If it finds it, then it moves to that cell. Otherwise, trying to avoid *Evil* it makes a random move.
- Oracle: this agent ‘foresees’ the cell where *Good* will be at the next step and if this cell is one of the neighbouring cells then it moves to that cell. Otherwise, it goes to the adjacent cell that, in the next iteration, will have the highest reward. Even though the ‘oracle’ has a sneaky advantage over the rest of the agents, it is not an ‘optimal’ agent, since it only foresees one-step movements, and this may be a bad policy occasionally.
- Q-learning: this is an off-the-shelf implementation of Q-learning, as explained in [17] and [14]. We use the description of cell contents as a state. Q-learning has two classical parameters: *learning rate* α and *discount factor* γ .

The choice of Q-learning as an example of a reinforcement learning algorithm is, of course, one of many possible choices, from a range of other RL algorithms from the same or different families. The reason is deliberate because we want a standard algorithm to be evaluated first, and, most especially, because we do not want to evaluate (at the moment) very specialised algorithms for ergodic environments or algorithms with better computational properties (e.g. delayed Q-learning [13] would be a better option if speed were an issue).

The parameters for Q-learning are $\alpha = 0.05$, $\gamma = 0.35$. The elements in the Q matrix are set to 2.0 initially (rewards range from -1 to 1 , but they are

normalised between 0 to 2 to always be positive in the Q matrix). The parameters have been chosen for the set of experiments in this paper by trying 20 consecutive values for α and γ between 0 and 1. These $20 \times 20 = 400$ combinations have been evaluated for 1,000 sessions each using random environments. So the parameters have been chosen to be optimal for the set of experiments included in this paper.

4 A Simple Example

Given the description of how environments are generated we will show how the previous agents perform in a single environment. The environment is based on the space in Fig. 1 and the following sequence for *Good* and *Evil*: 203210200. The number of steps (iterations) has been set to 10,000.

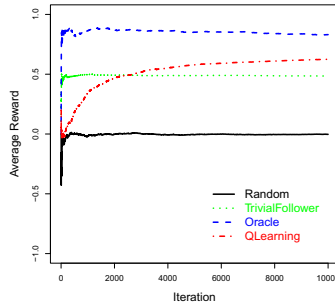


Fig. 2. Results for a simple environment over the 8-cell space in Fig. 1 and the sequences of actions that *Good* and *Evil* will follow: 203210200

The results of average reward for the several types of agents seen in fig. 2 show that initially the average reward has great fluctuations for the four types of agents. The random agent soon converges to its expected average reward, which is 0. The trivial follower is only able to score close to 0.5. In this environment and pattern, following *Good* is a good policy, but only to some extent. The oracle converges to a value around 0.83. As mentioned above, the oracle is near-optimal, and in many environments it will not reach a value of 1, but just a good value. Q-learning results in a slow convergence to a value of around 0.625. Although slow, in this case we see that it outperforms the trivial follower in the end.

Nonetheless, these results are only given for one particular environment. The following sections perform a battery of experiments which try to obtain some conclusions about the general behaviour of these agents.

5 Experiments

In this section we maintain the parameters and settings described in previous sections but now we average the results over many environments. In particular,

we choose 100 environments of 3 cells and 100 environments of 9 cells, which allow us to summarise the results for a range of cells between 3 and 9. Each environment has a random generation of the topology and a random generation of the sequence for *Good* and *Evil* as described above. The probability of stopping p_{stop} , which controls the size of the pattern, is set to $1/100$.

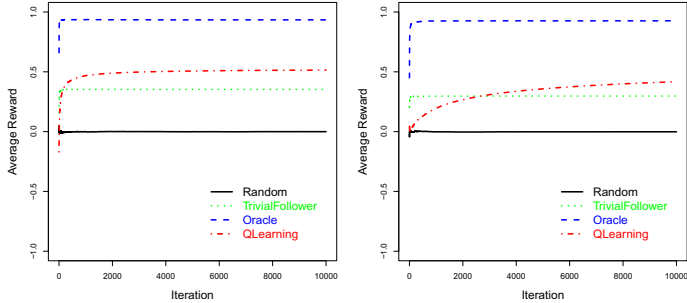


Fig. 3. Results for 100 environments. Left: 3 cells. Right: 9 cells.

The results are shown in Fig. 3. From these figures we get a general picture which is consistent with the single example shown above. Now we do not see fluctuations, since these are average results for 100 experiments each. We see that once contact is made with *Good*, then the policy is to follow it (somehow unsuccessfully for the trivial follower and very successfully for the ‘tricky’ oracle). Q-learning is able to surpass the trivial follower (by taking advantage of the pattern for *Good* and *Evil*) but just after a slow convergence, which is slower the higher the number of cells is, as expected.

We analyse the effect of complexity. In Fig. 4, we show the reward results of the four algorithms after 10,000 iterations compared to the ‘complexity’ of the environment. In order to approximate this complexity, we use the size of the compression of the description of the space (which obviously depends on the number of cells and actions), denoted by S , and the description of the pattern for *Good* and *Evil*, denoted by P . More formally, given an environment μ , we approximate its (Kolmogorov) complexity, denoted by K^{approx} , as follows:

$$K^{approx} = LZ(concat(S, P)) \times |P|$$

For instance, if the 8-cell 4-action space seen in Fig. 1 is coded by the following string $S = \text{“12+3----- | 12+++++3----- | 1-2-----3++ | 1-----2+++++3- | 12+3+++++ | 1-----23----- | 1+++++2-----3++ | 1----2+++3+”}$ and the pattern for *Good* and *Evil* is described by $P = \text{“203210200”}$, we concatenate both strings (total length 119) and compress the string (we use the ‘gzip’ method given by the *memCompress* function in R, a GNU project implementation of Lempel-Ziv coding). The length of the compressed string in this case is 60.

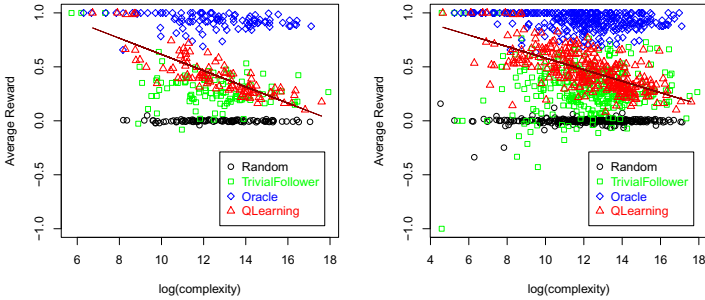


Fig. 4. Results after 10,000 iterations compared to the complexity of the environment. Left: 100 environments with 9 cells. Right: 300 environments with (3, 6, 9) of cells.

We see that the random agent, the oracle and the trivial follower are barely affected by complexity. On the contrary, Q-learning shows a significant decrease in performance as long as complexity is increased.

6 Discussion and Related Work

The experiments shown in previous sections portray a view about how an implementation of the intelligence test introduced in [6] using the environment class \mathcal{A} presented in [4] can be used to evaluate AI systems. Although the implementation has used several approximations and simplifications, we show that the theory works in capturing the essence of task complexity independently of the particular AI application or AI system to be evaluated.

Naturally, the application to the evaluation of RL systems is much more straightforward than other AI systems, since the test framework and reinforcement learning are based on the notion of interacting with an environment through observations, actions and rewards. Nonetheless, we think that most (if not all) tasks and areas in AI can be re-framed using this framework. In fact, in the previous tests, we have not made any single decision to favour or to specialise for any kind of learning algorithm or agent technology. Environment complexity is based on an approximation of Kolmogorov complexity, and not on an arbitrary set of tasks or problems. Consequently, our notion of complexity is not based on the idea of aliasing, Markov property, number of states, dimension, etc. Of course, all these issues are related to the notion of complexity, but we do not restrict to any subset of problems, or any notion of convergence, computational class, etc. Originally, the test just aims at using a Turing-complete environment generator, and it is the grading given by its (Kolmogorov) complexity which allows us to aggregate the performance obtained in each environment.

It is important to compare this approach to the traditional approach in artificial intelligence. There are some works on the evaluation of problem complexity

and also many AI competitions. For instance, Zatuchna and Bagnall [21] analyse mazes used in research in the last two decades, develop a specific measure of “complexity”, and try to determine which kind of learning agents behave best depending on the complexity, by also using a specific measure of performance (based on correctness, convergence, and memory). Extensive works, such as Zatuchna and Bagnall’s paper, are not frequent (because they require an enormous amount of work), but they are crucial for the real evaluation of progress in artificial intelligence. In our opinion, these evaluation works would be much more productive if they could be standardised under a grounded and common measurement of performance using the theory presented in [6]. In fact, what we have done here for reinforcement learning could also be restricted to ‘mazes’, using a proper environment class only representing mazes. Other areas such as multi-agent environments [18] could be adapted as well.

AI competitions are also a typical approach to evaluating progress in AI, such as, e.g., the AAAI General Game Playing Competition [2], or the RL-competition [19]. The latter is the closest approach to what we have done here. The RL-competition consists of several tasks for several domains. It is, in fact, several RL-competitions, one for each domain. Some environments are very specific. Others are a little bit more general, such as ‘polyathlon’, a set of ‘normalised’ classic and new reinforcement learning problems that look identical in terms of their task specification, so that the agent is not able to ‘identify’ which task it faces. This bottom-up approach is valuable, but we think that our top-down (theory-derived) approach is much more general and able to evaluate any kind of RL (or AI) agent without the risk of having systems specialised to it.

7 Conclusions

The goal of the paper was not to analyse some well-known properties of Q-learning (such as convergence, state overloading, etc.) – nor to designate a ‘winning’ algorithm. The goal of the paper, rather, was to show that a top-down (theory-derived) approach for evaluating AI agents can work in practice. We have used an implementation of [6] to evaluate Q-learning, as a typical off-the-shelf algorithm. We have seen the (inverse) relation of Q-learning performance with environment complexity. No restrictions about aliasing problems, partial observability, number of states, Markov properties, etc., are made here. As a direct application, several AI competitions and evaluation repositories could be defined using appropriate environment classes. The evolution of different algorithms with respect to the environment complexity would be one key feature to examine and a better indicator of progress in AI.

There is, of course, much work ahead. One clear area for future work is the evaluation of other reinforcement learning algorithms and the analysis of the parameters in all these algorithms (including Q-learning). In order to do this, we plan to integrate our system into the RL-glue architecture, so we could easily apply our tests to many existing RL algorithms already implemented for the RL-glue platform. One algorithm we want to evaluate soon is a Monte Carlo approximation to AIXI [16], which is showing impressive learning performance on

complex tasks and, interestingly, is based on ideas derived from Solomonoff prediction and Kolmogorov complexity. Another line for future work is to progress on a new version of the implementation of the test which could be more adherent to its full specification, by using better Turing-complete environment generators and better approximations for complexity. In this context, the implementation of the anytime version of the test in [6] (using the aggregation introduced in [5]) would also allow us to compare algorithms using efficiency as an important factor of the performance of the algorithms.

Finally, using our tests for humans and (non-human) animals will also be a very important source of information to see whether this top-down approach for measuring performance and intelligence can become mainstream in AI.

Acknowledgments. We thank the anonymous reviewers for their helpful comments. We also thank José Antonio Martín H. for helping us with several issues about the RL competition, RL-Glue and reinforcement learning in general. We are grateful for the funding from the Spanish MEC and MICINN for projects TIN2009-06078-E/TIN, Consolider-Ingenio CSD2007-00022 and TIN2010-21062-C02, for MEC FPU grant AP2006-02323, and Generalitat Valenciana for Prometeo/2008/051.

References

1. Dowe, D.L., Hajek, A.R.: A non-behavioural, computational extension to the Turing Test. In: Intl. Conf. on Computational Intelligence & multimedia applications (ICCIMA 1998), Gippsland, Australia, pp. 101–106 (1998)
2. Genesereth, M., Love, N., Pell, B.: General game playing: Overview of the AAAI competition. *AI Magazine* 26(2), 62 (2005)
3. Hernández-Orallo, J.: Beyond the Turing Test. *J. Logic, Language & Information* 9(4), 447–466 (2000)
4. Hernández-Orallo, J.: A (hopefully) non-biased universal environment class for measuring intelligence of biological and artificial systems. In: Hutter, M., et al. (eds.) 3rd Intl. Conf. on Artificial General Intelligence, Atlantis, pp. 182–183 (2010)
5. Hernández-Orallo, J.: On evaluating agent performance in a fixed period of time. In: Hutter, M., et al. (eds.) 3rd Intl. Conf. on Artificial General Intelligence, pp. 25–30. Atlantis Press (2010)
6. Hernández-Orallo, J., Dowe, D.L.: Measuring universal intelligence: Towards an anytime intelligence test. *Artificial Intelligence* 174(18), 1508–1539 (2010)
7. Legg, S., Hutter, M.: A universal measure of intelligence for artificial agents. Intl. Joint Conf. on Artificial Intelligence, IJCAI 19, 1509 (2005)
8. Legg, S., Hutter, M.: Universal intelligence: A definition of machine intelligence. *Minds and Machines* 17(4), 391–444 (2007)
9. Levin, L.A.: Universal sequential search problems. *Problems of Information Transmission* 9(3), 265–266 (1973)
10. Li, M., Vitányi, P.: An introduction to Kolmogorov complexity and its applications, 3rd edn. Springer-Verlag New York, Inc. (2008)
11. Sanghi, P., Dowe, D.L.: A computer program capable of passing IQ tests. In: Proc. 4th ICCS International Conference on Cognitive Science (ICCS 2003), Sydney, Australia, pp. 570–575 (2003)

12. Solomonoff, R.J.: A formal theory of inductive inference. Part I. *Information and Control* 7(1), 1–22 (1964)
13. Strehl, A.L., Li, L., Wiewiora, E., Langford, J., Littman, M.L.: PAC model-free reinforcement learning. In: *Proc. of the 23rd Intl. Conf. on Machine Learning, ICML 2006*, New York, pp. 881–888 (2006)
14. Sutton, R.S., Barto, A.G.: *Reinforcement learning: An introduction*. The MIT press (1998)
15. Turing, A.M.: Computing machinery and intelligence. *Mind* 59, 433–460 (1950)
16. Veness, J., Ng, K.S., Hutter, M., Silver, D.: Reinforcement learning via AIXI approximation. In: *Proc. 24th Conf. on Artificial Intelligence (AAAI 2010)*, pp. 605–611 (2010)
17. Watkins, C.J.C.H., Dayan, P.: Q-learning. *Machine learning* 8(3), 279–292 (1992)
18. Weyns, D., Parunak, H.V.D., Michel, F., Holvoet, T., Ferber, J.: Environments for multiagent systems state-of-the-art and research challenges. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) *E4MAS 2004. LNCS (LNAI)*, vol. 3374, pp. 1–47. Springer, Heidelberg (2005)
19. Whiteson, S., Tanner, B., White, A.: The Reinforcement Learning Competitions. *The AI magazine* 31(2), 81–94 (2010)
20. Woergoetter, F., Porr, B.: Reinforcement learning. *Scholarpedia* 3(3), 1448 (2008)
21. Zatuchna, Z., Bagnall, A.: Learning mazes with aliasing states: An LCS algorithm with associative perception. *Adaptive Behavior* 17(1), 28–57 (2009)

Evaluation of an Automated Mechanism for Generating New Regulations

Javier Morales^{1,2}, Maite López-Sánchez¹, and Marc Esteva²

¹ MAiA Dept., Universitat de Barcelona

{jmoralesmat, maite_lopez}@ub.edu

² Artificial Intelligence Research Institute (IIIA-CSIC)

marc@iiia.csic.es

Abstract. Humans usually use information about previous experiences to solve new problems. Following this principle, we propose an approach to enhance a multi-agent system by including an authority that generates new regulations whenever new conflicts arise. The authority uses a unsupervised version of classical Case-Based Reasoning to learn from previous similar situations and generate regulations that solve the new problem. The scenario used to illustrate and evaluate our proposal is a simulated traffic intersection where agents are traveling cars. A traffic authority observes the scenario and generates new regulations when collisions or heavy traffic are detected. At each simulation step, applicable regulations are evaluated in terms of their effectiveness and necessity in order to generate a set of regulations that, if followed, improve system performance. Empirical evaluation shows that the traffic authority succeeds in avoiding conflicting situations by automatically generating a reduced set of traffic rules.

1 Introduction

In any society, composed by humans or software agents, individuals continuously interact among them, and sometimes conflicts raise naturally. It has been proven that regulations are useful to enhance the running of societies by regulating individual's behavior and by solving conflictive situations. For instance, within juridical contexts, humans have developed Jurisprudence as the theory and philosophy of law, which tries to obtain a deeper understanding of general issues such as the nature of law, of legal reasoning, or of legal institutions [1]. Within it, Normative Jurisprudence tries to answer questions such as "*What sorts of acts should be punished?*". In the Anglo-American juridical system, when a new conflict arises it is usual to gather information about similar cases that were solved in the past to solve the current problem. Furthermore, when humans solve a new problem, sometimes they generate regulations in order to avoid that problem in the future. MAS societies, like human societies, can be enhanced by including specific regulations that promote a desired system's behavior. However, there are some key questions: "*When to generate new regulations?*", "*How to generate them?*" and "*How to know if the generated set of norms is correct?*".

¹ Jurisprudence definition extracted from Black's Law Dictionary:

<http://www.blackslawdictionary.com>

In a previous work [9] we answered these questions by proposing a computational mechanism that generates norms with the aim to improve the performance of the system. The aim of this paper is to present the resulting norm life cycle that defines the creation, maturing and establishment of sets of norms, as well as a more complete set of experiments.

2 Related Work

Research on norms in MAS is quite an active area. Campos et al. [4] have proposed norm adaptation methods to specific network scenarios; Boella and van der Torre have done relevant contributions [2] in norm characterization. Savarimuthu et al. [10], Griffiths and Luck [6], as well as Kota. et al. [8] work on norm emergence. Within this area, norm generation has been studied less frequently. Shoham and Tennenholtz [11] focus on norm synthesis by considering a state transition system: they explore the state-space enumeration and state it is NP-complete through a reduction to 3-SAT. Similarly, Hoek et al. [7] synthesize social laws as a model checking problem –again NP-Complete– that requires a complete action-based alternative transition system representation. Following this work, Agotnes and Wooldridge [1] extend the model by taking into account both the implementation costs of social laws and multiple (possibly conflicting) design objectives with different priorities. In this setting, the design of social laws becomes an optimization problem. Our approach does not explore the complete search space. Instead, we just explore a small portion of the search space by just expanding encountered conflictive states. Moreover, CBR has the advantage that, although cases are meant to cover the entire search space, they do not need to be exhaustive, since they can be representatives of a set of similar problems requiring similar solutions. Furthermore, our approach generates norms at run-time. This has the additional advantage of being able to regulate situations that may not be foreseeable at design-time. CBR allows the application to a wide range of domains, in particular to those where (i) experiences can be continuously gathered and evaluated, and where (ii) similar social situations require similar regulations (i.e., the continuity solution assumption). Within the MAS area Multi-Agent Reinforcement Learning [3] is quite widely used for individual agent learning. Nevertheless its usage is much more scarce for organizational centered approaches. Regarding Case Elicitation, in [2] an Unsupervised CBR system is used to solve new situations by learning from experience in a checkers game scenario;

Finally, regarding the traffic scenario, we highlight the MAS approach in [5], where an intersection agent assigns priorities to traveling cars according to pre-designed policies. They follow a control approach that implies a much tighter agent coordination than the one induced in our regulative approach.

3 The Traffic Scenario

The scenario represents an orthogonal two-road intersection discretized in a square grid of 20×20 cells. It is divided into five (disjoint) adjacent areas (see left of Fig. 1) covered by *monitor agents*. Cars are external agents with basic driving skills that enter into the scenario from four possible start points (dark/red points in left of Fig. 1), and

travel towards randomly chosen destinations (exit points, depicted in light/green in left of Fig. 1). Time is discrete (measured in ticks), and cars perform a single *action* $\in \{MoveForward, Stop, TurnLeft, TurnRight\}$ per tick. Cars move at constant *speed* of 1 cell per tick. More details about the scenario can be found in [9].

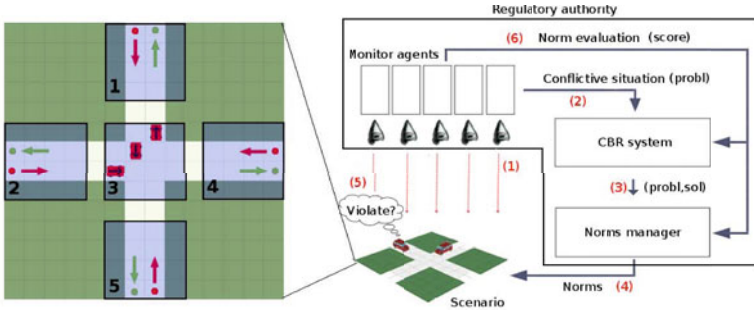


Fig. 1. Left: Zoom of the scenario. Right: Architecture of our system

4 Norm Life Cycle

We enhance our MAS with a regulatory authority (*traffic authority*) that, for new conflicts, generates norms by using the experience of previous similar cases. In the life cycle of a norm there are several stages.

Norm Generation Stage: As depicted in Figure 1 the *traffic authority* is permanently observing and gathering information from the scenario through five *monitor agents* (see label 1 in Fig. 1). When a new conflictive situation is detected, a description of it ($probl = \langle probl_{t-1}, probl_t \rangle$) is sent to the CBR system (2 in Fig. 1), where $probl_{t-1}$ (see Fig. 2a) is the situation previous to the conflict and $probl_t$ is the conflictive situation (see Fig. 2b). Then, CBR searches into the case base for cases that have a similar description. Cases are described as $Case = \langle probl, \{sol_i, score_i\} \rangle$, where $probl$ is the case description and $\{sol_i, score_i\}$ corresponds to a list of possible solutions, each one with its associated $score \in [0..1]$. Similarity between two cases A and B is computed as the inverse of the *distance* between their case descriptions. It is computed as the aggregation of distances of the cells of $probl$, comparing each cell in case A ($c_i^A \in probl_A$) with the corresponding cell in case B ($c_i^B \in probl_B$):

$$dist(probl_A, probl_B) = \sum_{i=1}^{nCells} dist(c_i^A, c_i^B)$$

Differences between two cells are considered to be 1 if their occupancy state is different, and 0 else (notice that this similarity function is commonly used for nominal attributes):

$$dist(c_i^A, c_i^B) = 1 \text{ if } state(c_i^A) \neq state(c_i^B) \text{ 0 else, where } state(c_i^k) = \{empty, car(heading, moving), collision\}$$

Since we may encounter symmetric cases, CBR applies rotations of α degrees to cases (where $\alpha \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$) while retrieving similar problems. When a case has been retrieved the system adapts its best solution to solve the new problem (see Fig. 2c). The adaptation process is done by rotating the solution the same α degrees than the retrieved case was rotated. Then, this new solution is added to the retrieved case. If the system lacks experience and no similar case was retrieved, a new pseudo-random solution is generated, assigning a stop obligation to one of collided cars.

Car agents may not be familiar with case syntax and so they may not be able to interpret case solutions. Hence, the *Norms Manager* translates case solutions into *norms* that agents can understand (3 in Figure 1). Norms are described as "IF *cond* THEN *obl(action)*", where *cond* is the condition for the norm to be applicable and *obl(action)* is the action to perform. The norm condition corresponds to the scope of this car, and the consequence of the norm is the obligation for that car to stop. Once a new norm is generated, its *score* is initially set to 0. Figure 2d depicts the resulting norm from the case example. Top part shows its graphical representation and bottom part its textual form. Generated norms are then communicated to the agents (see label 4 in Figure 1).

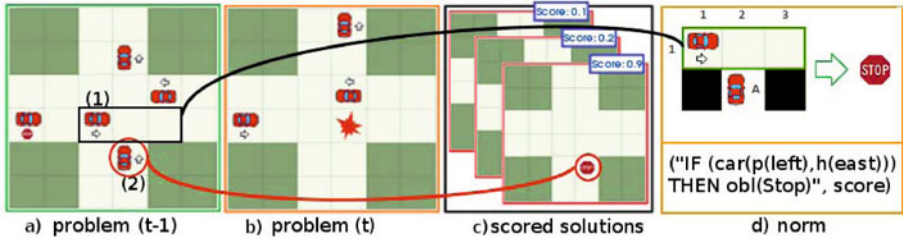


Fig. 2. Norm generation example: Case description in terms of a) $probl_{t-1}$ and b) $probl_t$; c) the set of generated solutions, and the resulting translated norm in d), where $p()$ is the position of the car and $h()$ is its heading

Norm Growth Stage: At each step cars use a *rule engine* to interpret which norms are applicable and decide whether to apply or violate them (5 in Figure 1). The *traffic institution* continuously gathers information about norm applications and violations so to evaluate norms (6 in Figure 1) and their associated case solutions in terms of a *score*. At each tick, the *traffic institution* detects which norms are applicable and evaluates them with respect to system goals considering their *effectiveness* and *necessity*. Specifically, norm applications are used to compute the *effectiveness* of a norm, checking whether a conflict arises (*ineffective norm*) or not (*effective norm*) after agents apply it. Norm violations are used to evaluate the *necessity* of a norm, checking whether a conflict arises (*necessary norm*) or not (*unnecessary norm*) after agents violate it. Therefore, norms are evaluated using the following formula:

$$\begin{aligned} eval &= effective - ineffective + necessary - unnecessary \\ &= K_E \times Ap_E - K_{-E} \times Ap_{-E} + K_N \times Viol_N - K_{-N} \times Viol_{-N} \end{aligned}$$

where Ap_E/Ap_{-E} are the number of applications that were effective/ineffective, and $Viol_N/Viol_{-N}$ denote the number of times a violation did/did not lead to a conflict.

Specifically, the value for each dimension is calculated by multiplying the number of occurrences of that kind by a constant factor K_i which is established by the designer and should be regarded as the importance given to that kind of situations.

Our current scenario considers two different goals, $G = \langle G_{cols}, G_{flTraff} \rangle$ which are directly related and contradictory. First goal (G_{cols}) is to avoid car collisions and second goal is to have fluid traffic ($G_{flTraff}$). On the one hand, optimizing G_{cols} requires cars to occasionally reduce speed or to stop in order to avoid collisions, causing heavier traffic and decreasing the performance of $G_{flTraff}$. On the other hand, optimizing $G_{flTraff}$ requires car not to stop, which decreases the performance of G_{cols} . Both goals are evaluated together in order to reach a trade-off between them. Since *ineffective* norms may cause collisions, the *effectiveness* of norms is directly related to the optimization of G_{cols} . *Unnecessary* norms cause unneeded stops and so heavier traffic, being prejudicial for $G_{flTraff}$. We can therefore instantiate the evaluation formula as:

$$eval = (K_E \times nCAppNoCol) - (K_{-E} \times nCAppCol) + \\ (K_N \times nCViolCol) - (K_{-N} \times nCViolNoCol)$$

where $nCAppNoCol$ is the number of cars that applied the norm and did not collide, $nCAppCol$ is the number of cars that applied the norm and collided, $nCViolCol$ is the number of cars that violated the norm and collided, and $nCViolNoCol$ is the number of cars that violated the norm and did not collide. Once $eval$ is computed, it is added to the history of evaluations of the norm, which comes down to be a window with $size = sz_{win}$. Finally, the score of the norm is computed by:

$$score = \frac{posEvals}{|negEvals| + posEvals}$$

where $posEvals$ is computed by adding all the values $eval \geq 0$ of the evaluation history, and $negEvals$ is computed by adding all the negative evaluation values ($eval < 0$) of the norm. Notice that with this method, the norm is evaluated in an iterative manner.

Norm Consolidation/Deactivation Stage: After norms have been evaluated a minimum number of times ($minEvals$), they are considered to have accumulated experience enough to determine if they must remain active or not. In case the $score$ value becomes under a certain *threshold*, the norm is deactivated and removed from the set of norms. Thus, it will not be applied any longer, unless it is generated again in another conflictive situation. Otherwise, if the norm remains active and its score is stable during the simulation, it is consolidated and considered as part of the optimal set of norms that optimize the running of the system.

5 Experiments

In order to evaluate our method and to compare its efficiency with standard coordination mechanisms (that is, with traffic lights) we have designed 4 different experiments. All experiments have been executed over the same simulator of the traffic scenario described in section 3. Since we evaluate norms, we just consider those collisions caused when norms are applied (instead of also including collisions coming from norm violations). The average of collisions is inversely proportional to the accomplishment of

G_{cols} . Similarly, the performance of $G_{flTraffic}$ is inversely proportional to the number of car stops. Therefore, these goals can be regarded as the minimization of the number of collisions and car stops respectively. Goals of our scenario (G_{cols} and $G_{flTraffic}$) are dependent and conflicting.

Due to the intrinsic randomness of the simulation, each experiment has been repeated 100 different times. Each simulation lasts 10000 *ticks*, and every 2 ticks, 3 new cars are added to the scenario. Thus, during simulations, the number that simultaneously populate the scenario can vary from 23 to 27. When norms are applicable, car agents have a probability $P(Violate) = 0.3$ of violating them. The size of the evaluations window is $size_{win} = 50$. Norms are deactivated when their score is under a $threshold = 0.3$ and they have been evaluated a minimum of 10 times ($minEvals = 10$) (see section 4).

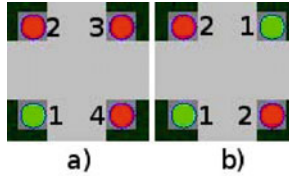


Fig. 3. Different configurations for traffic lights: a) 4 green light turns (1-East, 2-South, 3-West, 4-North) b) 2 green light turns (1-East & West, 2-North & South)

Experiment 1 uses our norm generation method to regulate the intersection. This experiment just considers G_{cols} in order to test if the system is able to accomplish one goal when no other factors are taken into account. For this aim, in this experiment constants are $K_E = 1$, $K_{-E} = 5$ and $K_N = K_{-N} = 0$. In order to compare our method with standard methods established by humans, in experiment 2 the scenario is regulated by traffic lights situated before entering the intersection, and there is no norm generation method. This approach is also used by K. Dressner and P. Stone in [5]. There are 4 lights, one for each lane. Traffic lights change their lights in 4 turns, as depicted in Figure 3.a. Thus, they give pass to the cars of one only lane at the same time.

Figure 4 depicts the results of both experiments 1 and 2. In experiment 1, using our norm generation method, the number of car stops is always lower than in experiment 2 (about 26 car stops per tick with traffic lights, and 4 car stops per tick with our method). This is due to the fact that, with traffic lights, cars are forced to stop following fixed patterns (i.e, time frequencies) regardless the actual traffic situation or traffic flow. On the other hand, with our approach norms describe situations and force cars to stop depending on a finer detail (the position of other cars). Thus, our method obtains a better performance for $G_{flTraffic}$. In experiment 2 the traffic lights configuration totally avoid collisions since traveling cars never find cars from another lane into the intersection. In experiment 1, collisions are eradicated from *tick* 550 on, optimizing the performance of G_{cols} . Since the system has one only goal G_{cols} , all norms that can eventually avoid collisions are included regardless the fact that they may be causing unneeded stops. Thus, the system rapidly converges to a stable set of 10 active norms that prevent collisions.

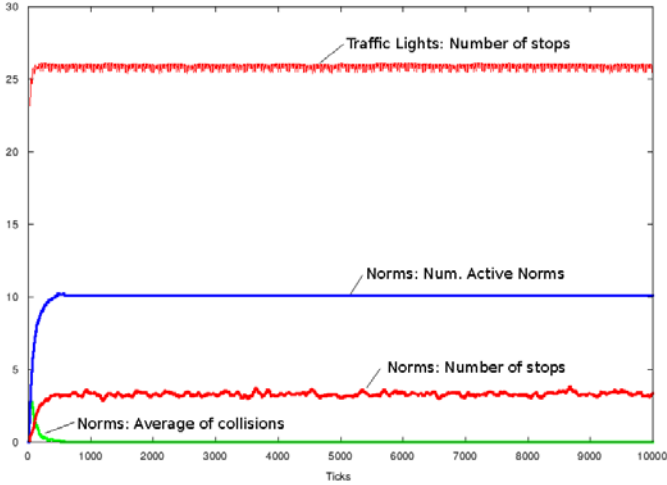


Fig. 4. Comparison of the results of experiments 1 and 2. Experiment 1: Norm generation with the goal of minimizing collisions. Experiment 2: Traffic light configuration of Figure 3a.

In order to study how norms can be generated when considering multiple conflicting goals, we have conducted a third experiment that applies our norm approach and considers both conflicting goals G_{cols} and $G_{flTraff}$. Specifically, constants are $K_E = 1$, $K_{\neg E} = 5$ and $K_N = 1$ and $K_{\neg N} = 2$. We compare the results of this experiment 3 with a fourth experiment, where an alternative setting of traffic lights improves the performance of fluid traffic. This is done by giving pass to two lanes simultaneously (see Figure 3b), with the associated penalty that collisions may happen into the intersection.

As depicted in Figure 5, the number of car stops per tick has decreased ($\simeq 19$) with respect to experiment 2. In experiment 3, using our method, the number of car stops has also decreased since now they are also part of the system goals ($G_{flTraff}$). Moreover, our method also optimizes fluid traffic much better than traffic lights, while collisions are relatively controlled ($\simeq 0.2$ per tick). In experiment 4, the average of collisions remains always higher than in experiment 3. Hence, our method optimizes both G_{cols} and $G_{flTraff}$ in a better way than traffic lights. Experiment 3 has conflicting goals, so the system is continuously activating and deactivating norms to find a trade-off between the performance of G_{cols} and $G_{flTraff}$. Hence, the system does not converge to a stable number of active norms. However, resulting norms partially fulfill both goals.

Two typical norms that always appear in all the performed simulations are:

- 1) IF (car(pos(left), heading(east))) THEN obl(Stop)
- 2) IF (car(pos(front), heading(north))) THEN obl(Stop)

Where $pos()$ is the *position* of a car and $heading()$ is its *heading*. Norm 1 corresponds to the *left-hand side priority* (see Fig. 2). In all performed simulations this norm (or its counterpart, the *right-hand side priority*) is always generated. This norm requires the car agent to stop if there is a car heading east to his left. In experiment 1, that uses

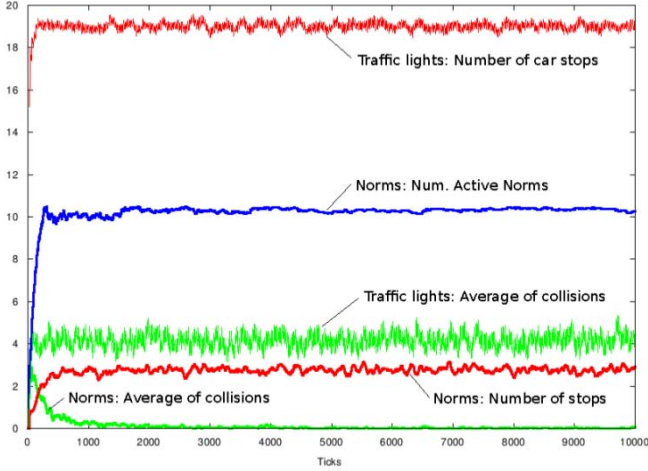


Fig. 5. Results for experiments 3 and 4. Experiment 3: Norm generation with both goals of minimizing collisions and to have fluid traffic. Experiment 4: Traffic light configuration of Figure 3b.

our norm generation method and takes into account only G_{cols} , this norm is generated and finally consolidated, but never deactivated. Thus, it is always part of the final set of norms. This is due to the fact that this norm is necessary, since its violations always lead to a collision. In experiment 3, that uses our approach and takes into account G_{cols} and $G_{flTraff}$, norm 1 is also always generated and consolidated. Since it is necessary, the norm also contributes to improve the performance of $G_{flTraff}$, being considered as part of the final set of norms that improve the running of the system.

Norm 2 can be regarded as a *security distance* norm. It is typically generated and applied in road areas out of the intersection (i.e., areas 1, 2, 4 and 5 in the left of Fig. 1). This norm requires the car agent to stop if there is a car in front of him with its same heading. Since it is preventive, sometimes cars violate it and collide, while some other times cars violate it and do not collide. In experiment 1, $G_{flTraff}$ is not taken into account and so this norm, that may seem unnecessary from the point of view of $G_{flTraff}$, is always included regardless the fact that it sometimes may cause unnecessary stops. Then, in this experiment it is generated and finally consolidated since it helps to minimize collisions (accomplishment of G_{cols}). In experiment 3 this norm, that goes against one of the goals ($G_{flTraff}$), is continuously being activated and deactivated because the system is trying to find a trade-off between the optimization of G_{cols} and $G_{flTraff}$. Specifically, this norm is always generated and occasionally it is deactivated because it is unnecessary from the point of view of $G_{flTraff}$. However, since the norm is necessary from the point of view G_{cols} , the norm is later generated again in another case and its life cycle starts again. As a consequence, collisions are not completely eradicated, but the number of car stops is reduced with respect to experiment 1.

6 Conclusions

This paper proposes a method to generate new regulations for multi-agent systems. Specifically, regulations are generated by a regulation authority using an unsupervised variation of Case Based Reasoning (CBR), when a conflictive situation arises. Generated norms are evaluated in an iterative manner in terms of their efficiency and necessity according to system goals. We thus claim that this innovative approach can be highly relevant for normative MASs, since, to the best of our knowledge, no general norm generation methods have been established yet that are capable to adapt the set of regulations during the execution of the system. Although norms are evaluated individually, their evaluation depends on the state reached each time they are applicable, and this state depends on all applicable norms. Applicable norms are then evaluated as a set of norms. If the application of a set of norms leads to a non-conflictive situation, the score of each norm would increase, while if their application leads to a conflictive situation, norms score would decrease.

This paper empirically evaluates our approach in the simulation of a simplified traffic scenario, where car collisions and traffic jams represent the conflictive situations and norms establish which circumstances a car must stop. Presented experiments compare our approach with standard traffic regulation methods like traffic lights. Results show how our method is capable to generate effective regulations taking into account single or multiple goals, improving the performance of system goals in a higher level than traffic lights.

Other scenarios requiring agent coordination —e.g. P2P networks, Robosoccer, etc.— may well benefit from our approach by avoiding conflictive situations —such as network saturation or teammate blocking in previous examples. As future work, we may consider the application our approach in other scenarios like these ones that have been just mentioned, and the application of other learning techniques such as Reinforcement Learning.

Acknowledgements. Work funded by EVE (TIN2009-14702-C02-01 / TIN2009-14702-C02-02) and CONSOLIDER AT (CSD2007-0022) projects and by the Generalitat de Catalunya under the grant 2005-SGR-00093. M. Esteva enjoys a Ramon y Cajal contract from the Spanish Government.

References

1. Agotnes, T., Wooldridge, M.: Optimal Social Laws. In: Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems, pp. 667–674 (2010)
2. Boella, G., van der Torre, L.: Regulative and constitutive norms in normative multiagent systems. In: Proceedings of KR 2004, pp. 255–265 (2004)
3. Busoniu, L., Babuska, R., de Schutter, B.: A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 38(2), 156–172 (2008), <http://dx.doi.org/10.1109/TSMCC.2007.913919>
4. Campos, J., López-Sánchez, M., Esteva, M.: Multi-Agent System adaptation in a Peer-to-Peer scenario. In: ACM Symposium on Applied Computing - Agreement Technologies Track, pp. 735–739 (2009)

5. Dresner, K., Stone, P.: A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research* 31, 591–656 (2008)
6. Griffiths, N., Luck, M.: Norm Emergence in Tag-Based Cooperation. In: 9th International Workshop on Coordination, Organization, Institutions and Norms in Multi-Agent Systems, pp. 79–86 (2010)
7. van der Hoek, W., Roberts, M., Wooldridge, M.: Social laws in alternating time: Effectiveness, feasibility, and synthesis. *Synthese* 1, 156 (2007)
8. Kota, R., Gibbins, N., Jennings, N.: Decentralised structural adaptation in agent organisations. In: AAMAS Workshop Organised Adaptation in MAS, pp. 54–71 (2008)
9. Morales, J., López-Sánchez, M., Esteva, M.: Using Experience to Generate New Regulations. In: Proceedings of the 22th International Joint Conference on Artificial Intelligence, IJCAI (2011)
10. Savarimuthu, B., Cranefield, S., Purvis, M., Purvis, M.: Role model based mechanism for norm emergence in artificial agent societies. In: Sichman, J.S., Padget, J., Ossowski, S., Noriega, P. (eds.) COIN 2007. LNCS (LNAI), vol. 4870, pp. 203–217. Springer, Heidelberg (2008)
11. Shoham, Y., Tennenholtz, M.: On social laws for artificial agent societies: off-line design. *Journal of Artificial Intelligence* 73(1-2), 231–252 (1995)

A Multi-agent System for Incident Management Solutions on IT Infrastructures

Elena Sánchez-Nielsen, Antonio Padrón-Ferrer, and Francisco Marreo-Estévez

Departamento de E.I.O. y Computación
Universidad de La Laguna
38271 La Laguna, Spain
enielsen@ull.es

Abstract. The use of multi-agent systems has contributed to the development of the theory and practice of successful solutions in diverse domains. In this paper, we present a solution that has been designed and implemented to support the real-world problem of incident management on IT infrastructures in order to restore a normal service operation as quickly as possible, ensuring that the best possible levels of service quality and availability are maintained. We take outstanding advantage of the following intrinsic characteristics of multi-agent systems in order to develop the proposed solution: (i) modeling the complex and distributed system through an agent architecture, (ii) negotiating as the communication process between different groups of agents in order to solve in an efficient way an incident when it is present, and (iii) automating the performance of the system through the analysis of incidents previously solved. A functionality scenario and example is illustrated as testing of the multi-agent system described.

Keywords: Multi-agent system, incident management, negotiating agents.

1 Introduction

Multi-agent systems are an important paradigm concerned with the analysis and development of sophisticated artificial intelligence problems with many existing and potential industrial and commercial applications. Examples of such application areas are: electronic commerce [1]; information management [2]; automated meeting scheduling [3]; electronic entertainment [4] and; healthcare services [5].

The goal of this paper is twofold. First, it is concerned to address the main reasons why the multi-agent systems can be considered one of the most interesting paradigms for the design of new applications related to the incident management problem in Information and Technology (IT) infrastructures and; second, to present a multi-agent system for the practical problem related to the incident management scenario, where agents communicate and negotiating in order to restore agreed service on the IT infrastructures as soon as possible with the least impact on either the business or the user and to respond to service requests.

The remainder of this paper is organized as follows. In section 2, the incident management problem is described, along with how potential solutions solve currently this problem. Section 3 describes the fundamentals of multi-agent systems and why this paradigm can be considered as an efficient solution to the challenging issues related to the incident management scenario. Based on this rationale, Section 4 illustrates the development and implementation of a multi-agent system to solve the incident management scenario. Concluding remarks are provided in Section 5.

2 Problem Description

According to the Information Technology Infrastructure Library (ITIL) [6], the *Incident Management* (IM) problem is focused on the Information and Technology (IT) Infrastructure scenarios, where incidents are the result of failures or errors in the IT infrastructure. An *incident* is any event which is not part of the standard operation of the service and which causes, or may cause, an interruption or a reduction of the quality of the service on the IT infrastructure. The main goal of IM is to restore a normal service operation as quickly as possible with the least possible impact on either the business or the user, at a cost-effective price, thus ensuring that the best possible levels of service quality and availability are maintained. This process is referred to as the *Incident Management Lifecycle*.

Normal service operation is defined here as service operation within a *Service Level Agreement* (SLA) [7], which establishes bounds about the average time to solve an incident. Inputs for IM mostly come from users. The outputs of the process are RFC's (Requests for Changes), resolved and closed incidents, management information and communication to the user. Hardware, software, networking, and different departments and organizations can be involved in the IM scenario. Template examples of some incidents related to the hardware service operations are e.g., "an automatic alert", or "printer not printing". Another incidents related to the software service operations are e.g., "the e-mail service is not available", "an application bug", or "disk-usage threshold exceeded".

The different components of the incident management lifecycle include: incident detection and recording, classification and initial support, investigation and diagnosis, resolution and recovery, incident closure, incident ownership, monitoring, and tracking and communication. Diverse critical success factors (CSFs) are used to measure the efficiency of the incident management process:

- **Maintaining IT Service Quality:** number of severity incidents (total and by category), number of other incidents (total and by category), number of incidents incorrectly categorized, number of incidents incorrectly escalated, number of incidents not closed/resolved, and number of incidents reopened.
- **Maintaining customer satisfaction:** average user survey score (total and by question category), and average queue time waiting for incident response.
- **Resolving incidents within established service times:** number of incidents logged, and average time to restore incidents.

Currently many existing commercial and free software applications for incident management are aimed at user level. Examples of such applications are: (1) Windows Help, and (2) Linux operating system. In addition, different commercial Web based Helpdesk software solutions are also available to offer several desirable properties for the management of organizations in order to recording incidents, new users, assignment of priority to solve incidents, and monitoring and tracking functionalities. Examples of such applications are: NetSupport DNA HelpDesk [8], Remedy Service Desk [9] and FootPrints [10]. All these applications provide interesting solutions to the incident management scenario. However, all these applications have not the capacity to offer the intrinsic properties of multi-agent systems. In the next section, we describe how the incident management applications can take outstanding advantage of the intrinsic characteristics of multi-agent systems to develop new solutions in the IM scenario.

3 Challenging Issues for Incident Management with Multi-agent Systems

A lot of work has been done in the last decade for spreading the use of multi-agent systems for the realization of smart software applications. Several technological specifications are the results of such work. Among them, the two main results to date are: (1) FIPA specifications [11], a set of specifications intended to support the interoperability between heterogeneous agent-based systems; and (2) an agent development framework, called JADE [12], that implements FIPA specifications and that supports interoperability between agents using consolidated programming languages e.g., Java. According to the incident management scenario described above, the main reasons to adopt a multi-agent system in this context are the following:

Natural View of Intelligent Systems: multi-agent systems offer a natural way to view, characterize, and design the incident management problem. In addition, multi-agent systems provide insights and understanding about interactions between agents, where coordination and negotiation are fundamental as they organize themselves into various groups with different rolls in order to achieve the appropriate actions.

Inherent Distribution: the incident management scenario is an inherent distributed process across multiple entities capable of intelligent coordination. This scenario is inherently distributed in the sense that the data and information to be processed related to the incidents management lifecycle: (1) arise at different computers ("spatial distribution"), (2) arise at different times ("temporal distribution") and, (3) are structured into clusters whose access and use requires different perceptual, effectual, and cognitive capabilities to solve the incident ("functional distribution").

Classification and Diagnosis: classification and diagnosis of incidents need the integration of different sources of data and the on-line collaboration of different users and/or technical experts. These features make multi-agent systems a reference model for their realization.

Speed-Up and Efficiency: agents can operate asynchronously and in parallel, and this can result in an increased overall speed. In addition, agents consume less network resources since they have a margin to take decisions in given situations related to incident management lifecycle.

Robustness and Reliability: the failure of one or several agents does not necessarily make the overall system unless, because other agents already available in the system may take over their part.

Scalability and Flexibility: the approach adopted can be increased by adding new agents with new functionalities (such as new technical agents to solve new incidents), and this does not affect the operability of the other agents.

Development and Reusability: individual agents can be developed separately by different specialists, the overall system can be tested and maintained more easily, and it may be possible to reconfigure and reuse agents in different situations.

4 Incident Management Application

As described above, multi-agent systems provide a natural way of system decomposition, organization, and abstraction, allowing that the inherent properties of a complex system can be reproduced in an agent-based system. That is, subsystems and subsystem components are mapped to agents and agent organizations; interactions between subsystems and subsystem components are mapped to communication, cooperation and negotiation mechanism; and relations between them are mapped to explicit mechanisms for representing organizational relationships. With this features in mind, the authors have treated the incident management lifecycle scenario through a multi-agent system framework. This framework has been implemented in JADE (Java Agent Development Framework) in compliance with the FIPA specifications for interoperable multi-agent systems, and the FIPA ACL as agent language communication. A domain ontology related to the incident management scenario was designed to represent the set of concepts and relationships between these concepts. The multi-agent system is illustrated in Fig. 4 and consists of seven distributed agent types.

4.1 Agent Types

Supervisor Agent (SVA)

This agent is a Graphical User Interface (GUI) through which the responsible of the IT organization or system administrator can configure the intrinsic characteristics of the incident management platform. *SVA* includes all the information related to technical experts grouped in thematic groups, users, and incidents (state of the incident lifecycle, priority assigned to incidents, incidents' categories, incident management policy assigned, what incidents are pending to be solved).

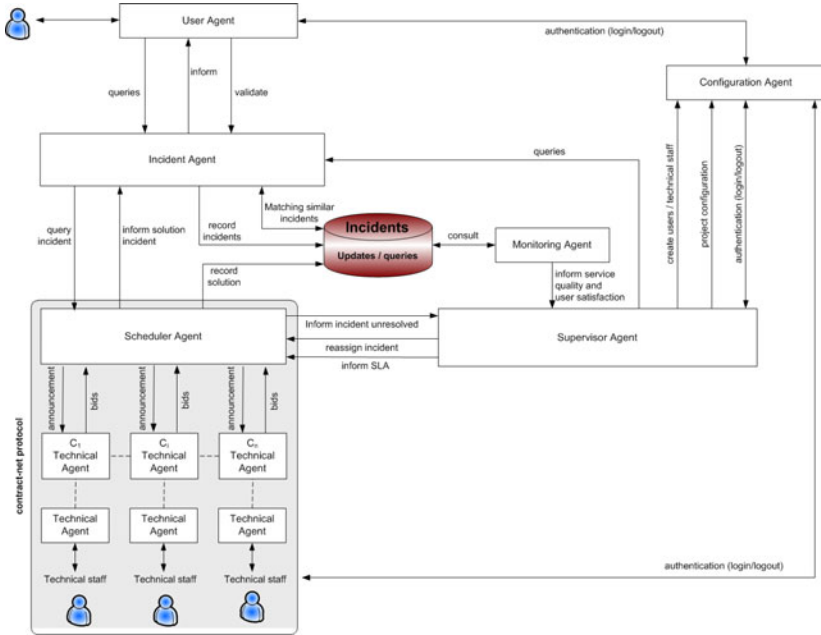


Fig. 1. Multi-agent system architecture

User Agent (UA)

A user agent is assigned to each user of the system. These agents are a GUI through which users can record incidents, monitoring and tracking the state of the different incidents. *UAs* include the entire user's personal information (name, address, phone number,...). From the GUI, users can look the priority, category and date of the recorded incident, solutions provided by the technical experts, who technical expert is solving the incident, and/or solutions to similar incidents solved, allowing users to validate or refuse these solutions. For controlling access to the application, a login/password system is provided through a configuration agent. The FIPA-Request protocol is used for authenticating and disconnecting of the system.

Technical Agents (TA)

The technical experts are grouped in different categories according to the taxonomy of incidents to be solved. The different groups are represented by the organizational structure C , where $C = \{C_1, C_2, \dots, C_n\}$. Examples of such technical category areas are: development applications support, office suites support, hardware and devices support, and communication infrastructures support. Each one of the different technical experts who belongs to the diverse categories of C is assigned to a *TA* agent, which keeps the schedule incident information, such

as pending incidents to be solved. Through a GUI, the technical experts view the assigned incidents, the SLA defined to the incident, severity of the incident, and the end-time to solve the incident. Through this friendly interface, the technical experts can also introduce annotations in the work agenda and provide the solution to the incident.

Incident Agent (IA)

This agent is the responsible of receiving the incidents generated by the distributed users through the corresponding *UA*. The different incidents are recorded in a database named *Incidents*. When asked by a *UA* for an incident solving, it looks in the Incidents database for computing all the similar incidents previously solved. In the case that similar incidents matched with the new incident recorded the corresponding solutions are sent from the *IA* to the corresponding *UA*. Alternatively, the incident is sent to the scheduler agent when no matching process has been computed.

Scheduler Agent (SA)

The negotiation process to assign the incident to the corresponding *TA* is applied via this agent. A Service Level Agreement (SLA) is assigned to each one of the different categories of incidents through the *SVA* by the responsible of the organization in order to restore a normal service as quickly as possible according to the severity incident. The SLA assigned defines the time slices required by the *TA* to solve the incident.

Monitoring Agent (MA)

The main role of *MA* is monitoring the global state of all the incidents of the system. These incidents are ranked according to the SLA. Through this agent, the responsible of the organization and/or the system administrator can supervise the efficiency of the incident management process related to the service quality, and user satisfaction allowing human responsible if is necessary to reassign incidents to other *TAs* through the *SA*.

Configuration Agent (CA)

This agent supervises the connection/disconnection of all the users, technical staff and responsible of the organizations to the system. The role of this agent also involves the configuration of user accounts, preventing the users from damaging the system, and informing *TAs* and *SVA* about changes in the project configuration.

4.2 System Functionality

The services offered by the agents described above inside the system are:

Recording Incidents: This service is offered by the *UA* to all users. From this *UA*, all the users can record a new incident and view the process incident

lifecycle. As a result, the users can be informed about the assignment and no assignment of their incidents, and who technical staff is solving the corresponding incidents.

Matching Incidents: when a new incident is recorded in the *Incident database*, the *Incident Agent* looks in the database for similar incidents previously solved. In order to formalize the text incident redacted by a user to a semantics understood by agents, the text incident is mapped to a quadruplet (four-tuple) representation: $\langle \textit{Object}, \textit{Negation}, \textit{Problem}, \textit{Feature} \rangle$, where the *Object* identifies the underlying cause of the incident; the *Problem* corresponds to the text verb of the incident redacted in infinitive form, the *Feature* represent a characteristic associated to the *Problem* and which is associated after the verb in the text sentence, and *Negation* indicates if this component text is present or not in the text incident redacted by the user. This representation structure has been implemented for Spanish language. For example, the three following incident text sentences: (i) "No me funciona el word", (ii) "El word no funciona", (iii) "El word no est funcionando" have the same semantic content and generate the following four-tuple representation: *Object*: word, *Problem*: funcionar, *Negation*: no, *Feature*: -. The ontology of the system includes a dictionary implemented by a graph data structure with the purpose of recording the different components of the four-tuple representation. Initially, when the system start from scratch, the technical staff is responsible of generating the four-tuple representation for the text incident. Subsequently, the system learns from the words introduced in the dictionary and then generate in an automated way, the four-tuple representation. Once the incident recorded in the system has been partitioned into the quadruplet structure by the *Incident Agent*, a matching process is computed in order to determine similar previously solved incidents. The matching process is based on comparing the new quadruplet structure with each one of the structures stored using a weighting algorithm. The different weights assigned to the four components of the quadruplet structured are based on the importance on detecting the underlying cause of the incident. Based on this rationale, the main weight is assigned to the *Object* component.

Assigning and Solving Incidents: the corresponding solution to an incident is sent from the *Incident Agent* to the *User Agent* when the matching process computes a similarity between the new incident and the previously incidents recorded on the *Incident database*. Otherwise, the *Scheduler Agent* searches the best technical staff to solve the incident. A negotiation technique based on a contract-net protocol [13] is used for searching the most appropriate *Technical Agent* to solve the incident. The FIPA-Contract Net Interaction Protocol is used as the value of the protocol parameter of the ACL message. This contracting mechanism consists of: (i) incident announcement by the *Scheduler Agent*; (ii) submission of bids from the Category C_i of *Technical Agents* in response to the announcement; and (iii) evaluation of the submitted bids by the *Scheduler Agent*, assigning the incident according to the most appropriate bids. The submission of bids from each category C_i of *Technical Agents* is computed according to the scheduling algorithms assigned by the responsible of the organization through

the *Supervisor Agent*. Four different scheduling algorithms are taken into consideration for configuring each one of the C_i categories of TA : (i) *Time-balancing policy*: the incident is assigned to the TA which has the biggest elapsed time slice of solving an incident; (ii) *Load-balancing policy*: the incident is assigned to the TA with the lowest number of incidents assigned; (iii) *Policy capacity load balancing*: this policy takes into the consideration the maximum number of incidents assigned to a TA (ratio) and the number of incidents assigned to the TA (n). The capacity load balancing assigned to an i^{th} TA is computed in the following way:

$$TA_i(\text{capacity load balancing}) = \frac{TA_i(\text{ratio})}{TA_i(n)} \quad (1)$$

and; (iv) *Dynamic balancing policy*: this policy combines all the different factors of the previous algorithm policies in the following way;

$$TA_i(\text{dynamic balancing policy}) = j * \frac{t}{T} + k * \left(1 - \frac{c}{R}\right) \quad (2)$$

Where t is the elapsed slice time of the TA_i of solving an incident; c is the number of incidents assigned to a TA_i ; T is the total time of all the TAs without an incident, and R corresponds to the TA_i ratio. The factors j and k are weight values, which are respectively initialized to the values 0.4 and 0.6.

Validating Incident Solutions: once a solution to an incident management is received by a user through the UA , he/she can validate the solution, and the incident is closed. Alternatively, the incident is assigned again when the solution proposed is refused, and the *Supervisor Agent* informs to the responsible of the organization about the unresolved incident.

Monitoring Incidents Lifecycle: the *Monitoring Agent* informs about the critical success factors used to measure the efficiency of the incident management process: (i) service quality: number of severity incidents (total and by category), number of other incidents (total and by category), number of incidents incorrectly categorized, number of incidents not closed/resolved, and number of incidents reopened; (ii) average queue time waiting for incident response; and (iii) number of incidents resolved within the established service time (SLA).

4.3 Testing the System

An experimental environment simulating a real-world IT infrastructure of a legislative assembly has been used to illustrate the main features of our application. For the experiment we have used a scenario with a responsible of the IT organization, 4 categories of technical agents, and 20 end users. A MySQL database was used to record and update incident data using real usage situations provided by the legislative assembly. The responsible of the organization when the system started from scratch used the CA and SVA to create user and technical

staff accounts, categories of incidents, severity of incidents, select scheduler algorithms to be used, and SLAs. The process of starting the system from scratch involved technical staff to record all the data of the incidents, such as recording the four-tuple representation and editing the work agenda with the solution procedure. From this, when similar incidents were recorded by users, they were matched with previously incidents through the *IA* in the *Incidents* database with an efficiency of 82 percent. Through the negotiation process, the incidents were assigned in an efficient way to the technical staff according to the scheduling algorithm configured. Dynamic balancing policy provided the best results regarding to restore the service quality from the viewpoint of technical staff and the responsible of the organization. New individual agents were easily developed and reconfigured during the testing of the system. All in all, the use of a multi-agent paradigm has much to offer with respect to traditional approaches allowing automate the services of the application and incorporate new agent capabilities and functionalities to the application in a flexible and scalar way when it is required through a cost-effective way. Fig. 2 shows the *UA* interface. Through this agent, the users recorded incidents and validated the solutions offered by the *TAs*. The established negotiations were between the *SA* and the *TAs*. When negotiations ended, the *SA* informed *UA* through the *IA* about the results of negotiations, that is, the solution to solve the incident.

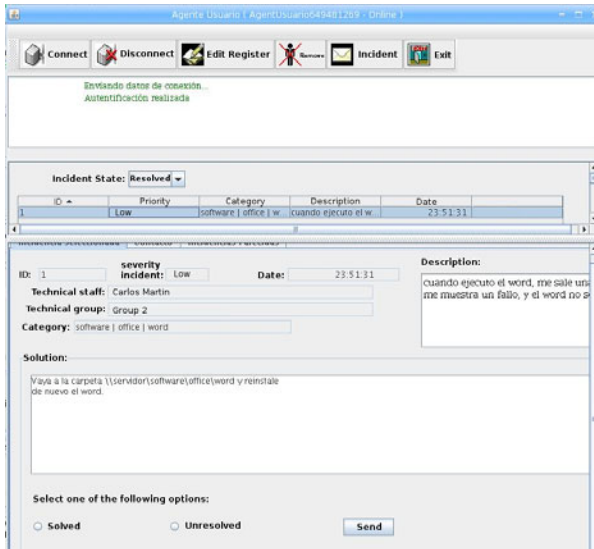


Fig. 2. *TA* and *UA* configuration: severity incident, category, description, date, technical staff assigned, and solution to be validated by the end user

5 Conclusions

The incident management problem is an important challenge in all the organizations which work with IT infrastructures in order to ensure that the best possible levels of service quality are maintained. In this paper, we show multi-agent systems potential to solve incident management problems. Modeling the system as a two level agent architecture which uses interface agents at upper level and negotiating agents at lower level has let us to carry out the phase of planning and solving an incident in an efficient, scalable, and automated way due to agent capabilities such as the communication, negotiation, and flexibility in relation to traditional and commercial approaches which have been focused on a centralized and no automated solution.

Acknowledgments. This work has been supported by Projects TIN2008-06570-C04-03 and PIL2210901.

References

1. Balachandran, B.M., Enkhsaikhan, M.: Developing Multi-agent E-Commerce Applications with JADE. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part III. LNCS (LNAI), vol. 4694, pp. 941–949. Springer, Heidelberg (2007)
2. Nguyen, N.T., Blazowski, A., Malowiecki, M.: A Multiagent System Aiding Information Retrieval in Internet Using Consensus Methods. In: Vojtáš, P., Bieliková, M., Charron-Bost, B., Sýkora, O. (eds.) SOFSEM 2005. LNCS, vol. 3381, pp. 399–402. Springer, Heidelberg (2005)
3. Zunino, A., Campo, M.: Chronos: A multi-agent system for distributed automatic meeting scheduling. *Expert Systems with Applications* 36(3), 7011–7018 (2009)
4. Berger, H., Dittenbach, M., Merkl, D., Bogdanovych, A., Simoff, S., Sierra, C.: Opening new dimensions for e-Tourism. *Journal Virtual Reality* 11(2) (2007)
5. Moreno, A., Valls, A., Isern, D., Sánchez, D.: Applying Agent Technology to Healthcare: The GruSMA Experience. *IEEE Intelligent Systems* 21(6), 63–67 (2006)
6. Official ITIL Website, <http://www.itil-officialsite.com/> (accessed May 2011)
7. Service Level Agreement and SLA Guide Website, <http://www.service-level-agreement.net/> (accessed May 2011)
8. NetSupport ServiceDesk Website, <http://www.netsupportservicedesk.com/index.asp> (accessed May 2011)
9. Remedy Service Desk Website, <http://www.artologik.com/es/HelpDesk.aspx> (accessed May 2011)
10. Footprints Website, <http://www.numarasoftware.es> (accessed May 2011)
11. FIPA Specifications, <http://www.fipa.org/specifications/index.html> (accessed May 2011)
12. JADE Website, <http://jade.tilab.com/> (accessed May 2011)
13. Smith, R., Davis, R.: The contract Net protocol: High level communication and control in a distributed problem solver. *IEEE Transactions on Computers* 29(12), 1104–1113 (1980)

Market Self-organization under Limited Information

Gregor Reich

Dept. of Economics, University of Basel, Switzerland
Gregor.Reich@unibas.ch

Abstract. The process of gradually finding an economic equilibrium, the so called tâtonnement process, is investigated in this paper. In contrast to classical general equilibrium modelling, where a central institution with perfect information about consumer preferences and production technologies (“Walrasian auctioneer”) organizes the economy, we simulate this process with learning consumer and producer agents, but no auctioneer. These agents lack perfect information on consumption preferences and are unable to explicitly optimize utility and profits. Rather, consumers base their consumption decision on past experience – formalized by reinforcement learning – whereas producers do regression learning to estimate aggregate consumer demand for profit maximization. Our results suggest that, even without perfect information or explicit optimization, it is possible for the economy to converge towards the analytically optimal state.

Keywords: Agent-based computational economics, market self-organisation, reinforcement learning, regression learning.

1 Introduction

Ever since the introduction of the concept of competitive market equilibrium by Leon Walras [14], economists have thought about the question how markets could eventually get to this state. And indeed, this is a reasonable question, since economic equilibrium is defined as a set of prices and endowments, such that every market participant is maximizing her profit or utility. Even more so, if one admits that the actually acting entities in the real world are people, possibly highly heterogeneous, that do not always decide based on solutions to complex optimization problems, but rather based on simple heuristics arising from experience and limited foresight.

To find a plausible explanation of how markets could eventually reach equilibrium, economic research focussed on finding different kinds of processes of price and trade quantity adoption. The first one was introduced by Walras himself, and is based on a central authority called the “Walrasian auctioneer”, who, by knowing all agents’ preferences and technologies, step by step adjusts the prices until all agents find themselves being in an optimal state. This process is obviously not very realistic for the decentralized markets found in reality, and moreover,

has been shown to be stable only under very restrictive assumptions. Other processes developed are the Edgeworth, the Hahn and the Fisher processes; but all of them still rely on very strict assumptions on agents' rationality and optimization abilities, and only the Fisher process does not involve centralized planning. See [7] for an overview of the processes and the stability issue.

A more recent development is the modelling of economies and its markets with Agent-based modelling techniques. These models have been applied widely in the field; see [13] for a review of the foundations and paradigms behind Agent-based computational economics, and [10] for a recent survey of the different applications. Recently, some contributions to the Agent-based computational economics literature brought equilibrium dynamics of markets and its comparison to theoretical benchmarks into focus [5,6]. Interestingly, they find that completely decentralized markets might organize, such that prices arising from bilateral trade converge towards the theoretically optimal (hence profit and utility maximizing) values under certain conditions. However, they still assume the agents to choose their actions by solving (constrained) optimization problems to maximize their utility, which is known to every single agent in its functional form.

Since we neither believe in peoples' perfect rationality and optimization abilities, nor in the assumption that people know "what makes them most happy" as a function of their actions (the utility function), we follow a different approach: In this paper, we model a consumption market where certain agents produce goods that other agents consume. We assume that everybody wants to make herself as "happy" as possible, namely by *implicitly* optimizing her utility from consumption by choosing combinations of goods that proved successful in the past. This is implemented with reinforcement learning, an algorithm whose application in economics has been pioneered by [4] and that has been used to study economic decision making and market dynamic analysis since then (see e. g. [9,16,8]); also, multi-agent reinforcement learning is used to study learning in stochastic games and its convergence and optimality properties (see e. g. [11,5], and [2] for a more general survey). At the same time, the agents producing consumption goods try to maximize their profits from product sales. The problem arising from imperfect information about consumer preferences is, that profit maximization involves knowing the trade off between rising a price to increase earnings per unit, and a decreasing demand for the good, since consumer buy less because they cannot afford them any more or because they switch to cheaper alternatives. This trade off, as a map between prices and demand quantities, is generally not known in this model quantitatively (how could it be, if not even consumers know their own preferences as a function of goods prices), but has to be estimated instead. We therefore use a regression learning algorithm (see [12]) to let producers learn the demand curve of their respective goods to maximize profits.

The question of this paper is whether our market model is able to organize itself without being lend a hand by a central authority. Furthermore, if this is the case, we want to check whether this stable state somehow corresponds to an economic equilibrium, where everybody has maximal attainable utility or profit.

Therefore, we will parametrize the model such that we are able to calculate the analytical solution to the so called social planner optimization problem, i. e. the solution of which makes everybody ending up optimally. We then simulate our model and compare the results to this benchmark.¹

2 The Model

On this artificial consumption goods markets, consumer and producers are repeatedly interacting in the following way: At the beginning of a trading period t , each producer j (a total of J) produces a fixed quantity of one unique consumption good and announces its price $p_{t,j}$. Prices and quantities are announced simultaneously for all goods and cannot be changed throughout the period. Once all prices have been announced, consumers (a total of I) repeatedly buy and immediately consume goods as long as (i) the goods are available (and hence not sold out) and (ii) their remaining income is greater or equal to the goods' respective prices. We assume that excess demand is (quantitatively) observable by the producers.² Consumers' aim is to distribute their spendable income among goods in a utility maximizing way, but they neither know the functional form of their utility function, nor have they any (direct) optimization abilities. Consequently, consumers have to rely on their experience of past consumption decisions, and they have learned how to behave optimally. We assume consumer income to be exogenously given. The consumer learning method is presented in detail in section 2.1.

The period is over once either all goods are sold out or all consumer income is spent; in the latter case, producers are not able to store their overproduction. By then, the next period starts with the producers price and supply quantity announcements. Each producer either keeps his price and supply strategy from the preceding period by announcing the same price and providing the same quantity of his good again, or, with some probability π_j , he revises his strategy and changes the price and the supply quantity. In the latter case, the producer chooses price and quantity such that his profits from expected sales are maximized; the estimation of expected sales given a certain price (the demand curve)

¹ Using the convergence towards some kind of equilibrium state as the only measure for a model's or a learning algorithm's quality has been criticized in both economics [3] and artificial intelligence [11]. We, too, believe that this is not where the analysis should end; rather, we think of it as a starting point for further investigations and future research, in order to identify the assumptions and conditions necessary to establish equilibrium.

² The problem of "truncated" observation of excess demand, hence the inability of producers to observe the true demand quantity given a certain price in case that demand is higher than actual supply, is present in reality of course. Since in this model, producers' price and supply quantity decisions directly depend on the consumer demand curve estimation solely relying on observed data, we are not able to relax this assumption without introducing a significant estimation bias, which in turn would affect the overall market outcome.

will be discussed in detail in section 2.2. The total number of producers J is constant over time: producers can neither go bankrupt, nor are there market entries. Since each producer has its unique good, there are J different goods for consumption. However, the relationship between goods, meaning their (dis-)similarities, are modelled solely by consumers' preferences. None of the agents knows anything about the others' preferences or production technologies. The only channel of information between agents are prices, aggregated demand and supply quantities.

2.1 Consumer Learning

Let $\mathcal{J}_{t,n}^i := \{j_{t,k}^i\}_{k=1}^n$, $j \in \{1, \dots, J\}$ be the sequence of goods consumed by agent i in period t , a total of n . Furthermore, let $u_{t,n}^i := U_{t,n}^i - U_{t,n-1}^i$ be the realized marginal utility of agent i in period t , where $U_{t,n}^i$ is the utility obtained from the consumption of $\mathcal{J}_{t,n}^i$. In words, $u_{t,n}^i$ is the increase of the overall utility level of one agent after having consumed one more good, namely good $j_{t,n}^i$. Then, the *propensity* of choosing good j for consumption next is

$$r_{t,n,j}^i = \begin{cases} r_{t,n-1,j}^i \bar{\gamma} + \alpha \frac{u_{t,n}^i}{p_{t,j}} & j_{t,n}^i = j \\ r_{t,n-1,j}^i & j_{t,n}^i \neq j \end{cases} \quad 0 < \bar{\gamma} < 1. \quad (1)$$

which is the sum of the marginal utility obtained from consuming $\mathcal{J}_{t,n}^i$, per unit of money spend, scaled by parameter α ; The factor $\bar{\gamma}$ implements forgetting within a period.³ The probability of agent i choosing good j is derived as follows: Let $\tilde{\mathcal{J}}_{t,n}^i$ be the set of goods that agent i can afford and that are still available on the market. Then,

$$pr_{t,n,j}^i := \begin{cases} r_{t,n,j}^i \cdot \left(\sum_{\substack{l=1 \\ l \in \tilde{\mathcal{J}}_{t,n}^i}}^J r_{t,n,l}^i \right)^{-1} & j \in \tilde{\mathcal{J}}_{t,n}^i \\ 0 & j \notin \tilde{\mathcal{J}}_{t,n}^i \end{cases} \quad (2)$$

Additionally, we set a lower bound on consumption choice probabilities $pr_{t,n,j}^i \geq \epsilon$ in order to ensure exploration of the whole action space.

It remains to define initial propensities at the beginning of each trading period, $r_{t,0,j}^i$. This is of great importance, since so far, consumers only maximize their utility within one period where goods prices stay fixed. However, if producers revise their price/quantity decisions at the beginning of a new period, consumers optimal response might change. In order to learn this, old experience has to be

³ We assume that *within* a consumption period, forgetting is applied to the past consumption experience for the actually chosen good only; experiments with forgetting applied to all possible actions returned – once calibrated properly – similar results, but slightly more variance.

given up (forgotten); but in order keep consumers continuing as before in case prices don't change, initial propensities of period $t + 1$ should be proportional to final choice probabilities at the end of period t . Consequently, we define

$$r_{t+1,0,j}^i = \tilde{\gamma} \cdot pr_{t,n,j}^i \quad (3)$$

where $\tilde{\gamma}$ controls the strength of initial propensities.

2.2 Producer Learning

Let $C^j(q^S)$ be the total cost function of producer j for output q^S . We assume that the cost function is known, and marginal costs MC^j (derivative of C^j with respect to q^S) can be derived. Consequently, profit maximization implies setting marginal costs equal to marginal revenues MR^j (derivative of total returns TR^j with respect to q^S)⁴. Since the computation of MR^j requires the knowledge of consumer demand curve (demand quantities for all possible prices), which is unknown by assumption, producers apply the following regression learning procedure:

Define $\tilde{T}_{t,j}$ to be the moving time window of aggregated consumer demand observation of length T_j , $\tilde{T}_{t,j} := \{u \in \{t, t-1, \dots, t-T_j\} \mid \mathbb{1}_{u-1} = 1\}$, where $\mathbb{1}_t$ is an index function taking on the value 1 if *any* producer changed its price at period t , and zero otherwise. Then, the linear demand function estimation model for time window $\tilde{T}_{t,j}$ and regression parameter vector β is

$$q_{\tilde{T}_{t,j},j}^D = \beta_0 + \beta_j p_{\tilde{T}_{t,j},j} + \beta_{-j} p_{\tilde{T}_{t,j},-j} + \varepsilon_j \quad (4)$$

where $q_{\tilde{T}_{t,j},j}^D$ is the history of demand observations for good j in $\tilde{T}_{t,j}$, and $p_{\tilde{T}_{t,j},j}$ are the corresponding prices; Index $-j$ stands for the set of all goods except j , $-j = \{k : k \in \{1, \dots, J\}, k \neq j\}$. Consequently, the predicted demand for $t + 1$ as a function of his and his competitors prices, $p_{t+1,j}$ and $\hat{p}_{t+1,-j}$, respectively, is

$$\hat{q}_{t+1,j}^D = \hat{\beta}_0 + \hat{\beta}_j p_{t+1,j} + \hat{\beta}_{-j} \hat{p}_{t+1,-j}. \quad (5)$$

Since the producers do not now the true future prices of their competitors $p_{t+1,-j}$, they will assume them to grow by their average growth rate within $\tilde{T}_{t,j}$. Since marginal costs and returns are in money rather than in quantity terms, we have to invert (5) to get the so called inverse demand function. Since at this point, the observed demand quantity $q_{\tilde{T}_{t,j},j}^D$, that was assumed to be a function of the price $p_{t+1,j}$, becomes in fact the decision variable dictating the price necessary to sell all units produced, we denote it as $q_{t+1,j}^S$ from now on. Finally, the inverse demand function writes as

$$p_{t+1,j} = \tilde{\beta}_0 + \tilde{\beta}_j q_{t+1,j}^S, \quad \tilde{\beta}_0 := -\frac{\hat{\beta}_0 + \hat{\beta}_{-j} \hat{p}_{t+1,-j}}{\hat{\beta}_j}, \quad \tilde{\beta}_j := \frac{1}{\hat{\beta}_j}. \quad (6)$$

⁴ This follows directly from the definition of total profits, which are equal to total returns less total cost, and setting its derivative to 0.

Taking the derivative of total returns yields marginal returns as a function of the intended supply quantity,

$$MR^j(q^S) = \tilde{\beta}_0 + 2\tilde{\beta}_j q^S. \quad (7)$$

Finally, producers set prices and quantities such that $MC^j(q^S) = MR^j(q^S)$ and (6) is fulfilled. The full market event sequence is summarized in Algorithm 1.

Algorithm 1. Artificial consumption goods market

```

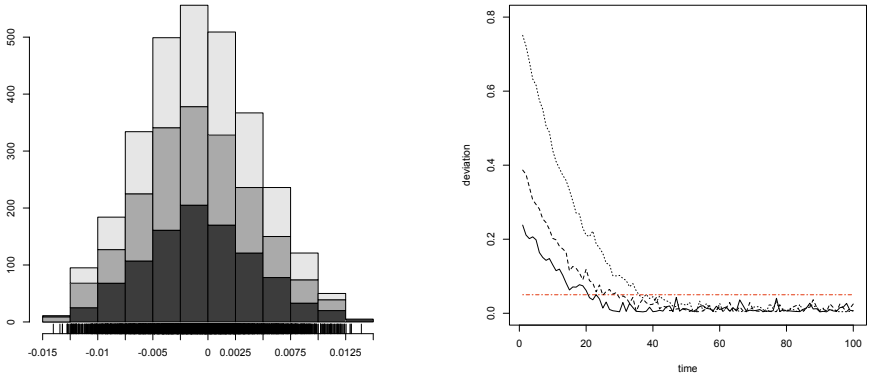
1: initialize algorithm
2: for number of iterations (index  $t$ ) do
3:   ## supply side:
4:   for all producers  $j$  do
5:     if  $j$  revises strategy then
6:       estimate demand function: (4)
7:       optimize profits: choose  $p_{t+1,j}$  and  $q_{t+1,j}^S$  s. t.  $MR^j = MC^j$ , (6)
8:     else
9:       set  $q_{t+1,j}^S = q_{t,j}^S$  and  $p_{t+1,j} = p_{t,j}$ 
10:    end if
11:  end for
12:  ## demand side:
13:  initialize propensities: (3)
14:  while (not all goods are sold out) and (not all income is spent) (index  $n$ ) do
15:    for all consumers  $i$  do
16:      choose affordable good  $j_{t+1,n}^i$  for consumption: (2)
17:      update propensities: (1)
18:    end for
19:  end while
20:  ## supply side:
21:  update demand history  $q_{T_{t+1,j},j}^D$ 
22: end for

```

3 Simulation Results and Validation

The presentation of the simulation results and their validation will be organized as follows: First, we validate the consumer learning procedure by fixing the price at some arbitrary level, and compare it to the benchmark. Then, we validate the producer regression learning procedure, independently of consumer learning, by letting consumers respond optimally as under perfect information. Last, we put things together and simulate and validate the whole model.

We parametrize the model as follows: There are two goods offered on the market, and consumers' utility function is $U(q_1, q_2) = 1.2 \cdot q_1^\sigma + q_2^\sigma$ with $\sigma = 0.7$ in order to make the goods non-perfect substitutes. For the consumer validation, prices and income s are fixed at $p_1 = 1.4$, $p_2 = 1$ and $s = 100$. Forgetting and step size are $\bar{\gamma} = .9999$, $\tilde{\gamma} = 100$, and $\alpha = 1$. Each simulation involves 100 agents and 100 consumption periods.



(a) Terminal relative deviation from benchmark.

(b) Relative deviation from benchmark over time (slash-dotted: 5% line).

Fig. 1. Simulation and validation results for the consumer learning algorithm for three values of initial propensities

The simulation results for the consumer validation are reported in Fig. [1\(a\)](#), which depicts a histogram of the terminal relative deviation of the consumption quantity (good 1) from its benchmark solution⁵. We see that the distribution is unbiased, symmetrical, and almost all observations lie within 1.5 percent of deviation from the benchmark, which we think is very satisfying. For the simulations, we used three different values of initial propensities $r_{0,0,j}^i$, with 1000 runs each. The colour further dividing the histogram bars represent the share of observations starting from each particular initial propensity value, and we conclude that they do not affect the final simulation outcome. Figure [1\(b\)](#) shows the same deviation measure of three particular simulation runs (again starting from three different initial propensities) over time.

We now turn to the discussion of the producer learning algorithm. As indicated, we first run it based on the optimal consumer response, in order to validate this particular algorithm and rule out effects arising from the interoperation with the consumer learning. We assume the cost function to be quadratic and parametrized such that the minimum of average costs is at 50000, and the minimum of MC^j is at 32000. The observation time window length is $T_j = 2000$, and producers change their prices and quantities every 100 periods in average. There are 1500 consumers, each with 16000 units of income; to control for nominal effects of the much higher equilibrium prices compared to the consumer validation example, we set $\alpha = 400$. Figure [2\(a\)](#) presents the results as a two-dimensional histogram of the joint terminal state distribution of the demand curve slope

⁵ Figures [1\(a\)](#) is trimmed at the one percent level.

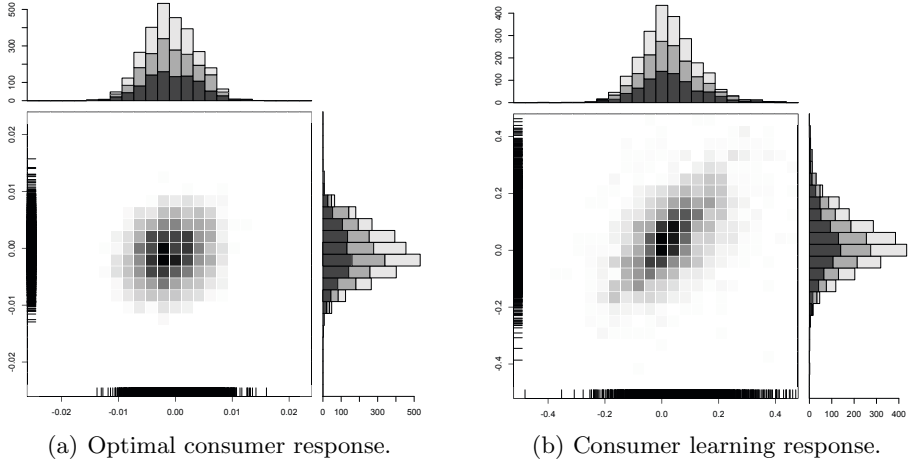


Fig. 2. Terminal relative deviation from benchmark for the producer regression learning algorithm for three values of initial propensities

estimate (relative deviation from benchmark value)⁶; above and beside it, the marginal distributions are plotted, showing the impact of three different starting points for the estimation, similar to what we did in Fig. 1(a). For each of them, 1000 simulations of 25000 periods length were carried out. We conclude that the algorithm is nearly unbiased, starting point independent, and that it has very low variance. Additionally, the distribution seems to have no correlation, so the direction of deviation from the benchmark of one producer is independent of the other one's; hence, we think of it as being noise only.

Turning to the full market simulation, we now simulate both the consumer and the producer as learning agents; the results are presented in Fig. 2(b). Again, the market seems to converge towards equilibrium (unbiased), but this time, the variance of the terminal states of the system is much higher⁷. Moreover, there is significant correlation in demand curve estimation error. We think that this is due to the fact that learning consumers take longer to adapt to changes in consumption good prices, and hence producers take longer to learn that it is optimal for them to be closer to the benchmark than their competitors.

However, the convergence of the full market simulation towards the equilibrium comes at no surprise: The configuration of utility and cost functions we use is such that a unique Nash equilibrium is established. Moreover, this Nash equilibrium is attractive. In the production context, this means that (i) in equilibrium, no producer can increase its profits given the other producers strategies

⁶ Figure 2(a) is trimmed at the eight percent level.

⁷ About 25% of the simulation runs of the full market simulation ended early, because the system got destabilized and moved towards states where demand curve estimation was impossible. These observations are not included in Fig. 2(b).

and therefore all players remain passive, and (ii) out of equilibrium, for the producer with the highest deviation from the benchmark (in terms of profits), it is always more profitable to correct his strategy towards it, and no producer can increase his profits by moving further away from the benchmark than the producer with the highest deviation. Consequently, if consumer learning for fixed prices yields nearly optimal consumption bundles, and if producer approximate the true demand curve correctly for optimal consumer response (as it is the case for the utility function in use), the market will finally converge. Of course, further research is needed for utility configurations with multiple or lacking Nash equilibria, possibly revealing periodic or even chaotic behaviour.

Checking our model for sensitivity to changes in parameter values, we find that with respect to the producer learning, the model seems to be robust as long as (i) the frequency of producers changing prices and output quantities π_j , in relation to the demand observation window length T_j , provides high enough a number of observations for the estimation procedure, and (ii) the consumer agents have enough time to adapt to the new price regime and therefore respond in a way that reflects their actual preferences. For the consumer learning algorithm, forgetting (in relation to the step size parameter α and the coefficients of the underlying utility function) needs to be calibrated properly. We find that our specific setting is robust with respect to the ratio of goods prices, as long as α is in the order of magnitude of the prices.

4 Conclusions and Outlook

We have presented an Agent-based model of a market for heterogeneous consumption goods, where consumers do not know their utility function, but can only learn to maximize it from past consumption experience; at the same time, producers estimate the demand curve for their goods from historical data in order to maximize their profits. Our simulation results indicate that the market is self-organizing, and moreover, establishes an optimal state from the viewpoint of profit and utility maximization, without the assumption of perfect information about agents' preferences and technologies, and without a central authority that organizes the market. However, relating our result to reality, we have to conclude that even if the model would be correct, and reality would be just one draw from the distribution in Fig. 2(b), variance is still too high to conclude that the real world is likely to be at its profit and utility maximizing equilibrium.

Further research will on the one hand investigate the market's convergence (or periodicity) behaviour for different utility functions, since we expect the existence and uniqueness of the Nash equilibrium of the current configuration to be the driving force behind our results. On the other hand, we will ask whether the market is also stable, and moreover, whether the "law of one price" is established in case that the goods are perfect substitutes, meaning that they are perfectly interchangeable for agents without any loss in utility. In the existing literature, this can only be achieved by assuming that goods prices are no common knowledge among agents, an assumption we think to be unrealistic, too.

Acknowledgements. I thank Dietmar Maringer for many helpful discussions. In addition, I thank three anonymous reviewers for helpful comments. Financial support of the WWZ Forum under grant D-131 is gratefully acknowledged.

References

1. Bowling, M., Veloso, M.: Multiagent learning using a variable learning rate. *Artif. Intell.* 136(2), 215–250 (2002)
2. Busoniu, L., Babuska, R., De Schutter, B.: A comprehensive survey of multiagent reinforcement learning. *IEEE T. Syst. Man. Cy. C.* 38(2), 156–172 (2008)
3. Colander, D., Rothschild, C.: Sins of the sons of samuelson: Vision, pedagogy, and the zig-zag windings of complex dynamics. *J. Econ. Behav. Organ.* 74(3), 277–290 (2010)
4. Erev, I., Roth, A.E.: Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *Am. Econ. Rev.* 88(4), 848–881 (1998)
5. Gintis, H.: The dynamics of general equilibrium. *Econ. J.* 117(523), 1280–1309 (2007)
6. Gintis, H.: The dynamics of generalized market exchange. Santa Fe Institute Working Paper (2011)
7. Hsieh, C.Y., Mangum, S.L.: A search for synthesis in economic theory. M.E. Sharpe, New York (1986)
8. Kirman, A., Vriend, N.: Evolving market structure: An ace model of price dispersion and loyalty. *J. Econ. Dyn. Control* 25(3-4), 459–502 (2001)
9. Rieskamp, J., Busemeyer, J., Laine, T.: How do people learn to allocate resources? comparing two learning theories. *J. Exp. Psychol. Learn.* 29(6), 1066–1081 (2003)
10. Rouchier, J.: Agent-based simulation as a useful tool for the study of markets. Technical Report 8, GREQAM (2008)
11. Shoham, Y., Powers, R., Grenager, T.: If multi-agent learning is the answer, what is the question? *Artif. Intell.* 171(7), 365–377 (2007)
12. Spall, J.: Introduction to stochastic search and optimization: estimation, simulation, and control. Wiley Interscience, Hoboken (2003)
13. Tesfatsion, L., Judd, K.L.: Handbook of Computational Economics, vol. 2: Agent-Based Computational Economics. North-Holland, Amsterdam (2006)
14. Walras, L.: Elements of pure economics, or, The theory of social wealth. Allen and Unwin, London (1874/1954)
15. Wang, X., Sandholm, T.: Reinforcement learning to play an optimal nash equilibrium in team markov games. In: Becker, S., Thrun, S., Obermayer, K. (eds.) *Advances in Neural Information Processing Systems*, vol. 15, pp. 1571–1578. MIT Press, Cambridge (2003)
16. Weisbuch, G., Kirman, A., Herreiner, D.: Market organisation and trading relationships. *Econ. J.* 110(463), 411–436 (2000)

Solving Sequential Mixed Auctions with Integer Programming

Boris Mikhaylov, Jesus Cerquides, and Juan A. Rodriguez-Aguilar

Artificial Intelligence Research Institute,
IIIA Spanish National Research Council, CSIC
{boris,cerquide,jar}@iia.csic.es

Abstract. Mixed multi-unit combinatorial auctions (MMUCAs) offer a high potential to be employed for the automated assembly of supply chains of agents. However, in order for mixed auctions to be effectively applied to supply chain formation, we must ensure computational tractability and reduce bidders' uncertainty. With this aim, we introduce Sequential Mixed Auctions (SMAs), a novel auction model conceived to help bidders collaboratively discover supply chain structures. Thus, an SMA allows bidders progressively build a supply chain structure through successive auction rounds.

1 Introduction

According to [7], “Supply Chain Formation (SCF) is the process of determining the participants in a supply chain, who will exchange what with whom, and the terms of the exchanges”. Combinatorial Auctions (CAs) [2] are a negotiation mechanism well suited to deal with complementarities among the goods at trade. Since production technologies often have to deal with strong complementarities, SCF automation appears as a very promising application area for CAs. However, whilst in CAs the complementarities can be simply represented as relationships among goods, in SCF the complementarities involve not only goods, but also *transformations* (production relationships) along several levels of the supply chain.

The first attempt to deal with the SCF problem by means of CAs was done by Walsh et al. in [7]. Later on, mixed multi-unit combinatorial auctions (MMUCAs), a generalization of the standard model of CAs, are introduced in [1]. Rather than negotiating over goods, in MMUCAs the auctioneer and the bidders can negotiate over *transformations*, each one characterized by a set of input goods and a set of output goods. A bidder offering a transformation is willing to produce its output goods after having received its input goods along with the payment specified in the bid. While in standard CAs, a solution to the winner determination problem (WDP) is a set of atomic bids to accept, in MMUCAs, the *order* in which the auctioneer “uses” the accepted transformations matters. Thus, a *solution* to the WDP is a *sequence of transformations*. For instance, if bidder *Joe* offers to make dough if provided with butter and eggs, and bidder

Lou offers to bake a cake if provided with enough dough, the auctioneer can accept both bids whenever he uses Joe's transformation before Lou's to obtain cakes. Unfortunately, the MMUCA WDP has been proved to be NP-complete [1]. Although reasonably fast solvers have been introduced [4], MMUCA still turns out to be impractical in real-world procurement scenarios. Furthermore, a bidder in MMUCA only knows the desired outcome of the supply chain and the current stock goods. Hence, it is difficult, specially for providers placed in the intermediate levels of the supply chain, to decide what to bid for. Therefore, in order for mixed auctions to be effectively applied to SCF, we must ensure computational tractability and reduce bidders' uncertainty. With this aim, we introduce Sequential Mixed Auctions (SMAs), a novel auction model conceived to help bidders collaboratively discover supply chain structures.

SMAs propose to solve a SCF problem by means of a sequence of auctions. The first auctioning round starts with the desired outcome of the supply chain as requested goods and the stock goods as available goods. During the first auction, bidders are only allowed to bid for transformations that either (i) produce goods in the set of requested goods or (ii) consume goods from the available goods. After selecting the best set of transformations, the auctioneer updates the set of requested and available goods after the execution of these transformations and then it will start a new auction. The process continues till no bids can be found that improve the supply chain. Notice that each auction in the sequence involves only a small part of the whole supply chain, instead of the whole one as MMUCAs do. Thus, auctions in an SMA are much less computationally demanding than a MMUCA. Moreover, the incremental nature of an SMA provides its participants with valuable information at the end of each auction round to guide their bidding.

The paper is organised as follows. Section 2 provides some background of mixed auctions, whereas section 3 formally states the WDP and section 4 details a mixed integer program that solves it. Finally, section 5 concludes.

2 Background: Mixed Auctions

Next we summarise the work in [1], which introduces mixed auctions (MMUCAs) as a generalisation of CAs and discusses the issues of bidding and winner determination.

Let G be the finite set of all the types of goods. A *transformation* is a pair of multi-sets over G : $(\mathcal{I}, \mathcal{O}) \in \mathbb{N}^G \times \mathbb{N}^G$. An agent offering the transformation $(\mathcal{I}, \mathcal{O})$ declares that it can deliver \mathcal{O} after having received \mathcal{I} . Bidders can offer any number of such transformations, including several copies of the same transformation. That is, agents negotiate over *multi-sets of transformations* $\mathcal{D} \in \mathbb{N}^{(\mathbb{N}^G \times \mathbb{N}^G)}$. For example, $\{(\{\}, \{a\}), (\{b\}, \{c\})\}$ means that the agent in question can deliver a (no input required) and that it can deliver c if provided with b .

Since agents negotiate over bundles of transformations, a *valuation* $v : \mathbb{N}^{(\mathbb{N}^G \times \mathbb{N}^G)} \rightarrow \mathbb{R}$ is a mapping from multi-sets of transformations to real numbers. Intuitively, $v(\mathcal{D}) = p$ means that the agent equipped with valuation v is willing

to make a payment of p for being allocated all the transformations in \mathcal{D} (in case p is negative, this means that the agent will accept the deal if it *receives* an amount of $|p|$). For instance, valuation $v(\{\{\textit{line}, \textit{ring}, \textit{head}, 6 \cdot \textit{screws}, \textit{screwdriver}\}, \{\textit{cylinder}, \textit{screwdriver}\}\}) = -10$ means that some agent can assemble a cylinder for 10€ when provided with a (cylinder) line, a (cylinder) ring, a (cylinder) head, six screws, and a screwdriver, and returns the screwdriver once done.¹

An *atomic bid* $b = (\{(\mathcal{I}^1, \mathcal{O}^1), \dots, (\mathcal{I}^n, \mathcal{O}^n)\}, p, \beta)$ specifies a finite multi-set of finite transformations, a price p and the bidder β . A *bidding language* allows a bidder to encode choices between alternative bids and the like [6]. Informally, an OR-combination of several bids means that the bidder would be happy to accept any number of the sub-bids specified, if paid the sum of the associated prices. An XOR-combination of bids expresses that the bidder is prepared to accept at most one of them. The XOR-language is known to be fully expressive for MMUCAs [1]. Bids in MMUCAs are composed of transformations. Each transformation expresses either an offer to buy, to sell, or to transform some good(s) into (an)other good(s). Thus, transformations are the building blocks composing bids. We can classify the types of transformations over which agents bid as follows:

- 1. Output Transformations** are those with no input good(s). Thus, an O-transformation represents a bidder’s offer to sell some good(s).
- 2. Input Transformations** are those with no output good(s). Thus, an I-transformation represents a bidder’s offer to buy some good(s).
- 3. Input-Output Transformations** are those whose input and output good(s) are not empty. An IO-transformation stands for a bidder’s offer to deliver some good(s) after receiving some other good(s): *I can deliver \mathcal{O} after having received \mathcal{I}* . They can model a wide range of different processes in real-world situations (e.g. assembly, transformation, or exchange).

The *input* to the WDP consists of a complex bid expression for each bidder, a multi-set \mathcal{U}_{in} of (stock) goods the auctioneer holds to begin with, and a multi-set \mathcal{U}_{out} of (required) goods the auctioneer expects to end up with. In standard CAs, a solution to the WDP is a set of atomic bids to accept. As to MMUCAs, the *order* in which the auctioneer “uses” the accepted transformations matters. For instance, if the auctioneer holds a to begin with, then checking whether accepting the two bids $Bid_1 = (\{\{a\}, \{b\}\}, 10, id_1)$ and $Bid_2 = (\{\{b\}, \{c\}\}, 20, id_2)$ is feasible involves realizing that we have to use Bid_1 before Bid_2 . Thus, a *valid solution* to the WDP will be a *sequence of transformations* that satisfies:

- (1) *Bidder constraints*: The multi-set of transformations in the sequence has to *respect the bids* submitted by the bidders. This is a standard requirement. For instance, if a bidder submits an XOR-combination of transformations, at most one of them may be accepted. With no transformation free disposal, if a bidder submits an offer over a bundle of *transformations*, all of them must be employed in the transformation sequence, whereas in the case of transformation

¹ We use *6 · screws* as a shorthand to represent six identical elements in the multi-set.

free disposal any number of the transformations in the bundle can be included into the solution sequence, but the price to be paid is the total price of the bid.

(2) *Auctioneer constraints*: The sequence of transformations has to be *implementable*: (a) check that \mathcal{U}_{in} is a superset of the input set of the first transformation; (b) then update the set of goods held by the auctioneer after each transformation and check that it is a superset of the input set of the next transformation; (c) finally check that the set of items held by the auctioneer in the end is a superset (the same set in the case of no good free disposal) of \mathcal{U}_{out} .

An *optimal* solution is a valid solution that maximizes the sum of prices associated with the atomic bids selected.

The WDP for MMUCAs is a complex computational problem. In fact, one of the fundamental issues limiting the applicability of MMUCAs to real-world scenarios is the computational complexity of the WDP, which is proved in [11] to be \mathcal{NP} -complete. Although [4] introduces an integer program to efficiently solve the WDP that drastically outperforms the original IP described in [11], the computational complexity impedes scalability. The next section introduces a new mixed auction model that allows to tame complexity while reducing bidders' uncertainty.

3 Sequential Mixed Auctions

An SMA proposes to solve the SCF problem by means of a sequence of auctions. The first auction in the sequence starts with the desired outcome of the supply chain as requested goods and the stocked goods as available goods. During the first auction, bidders are only allowed to bid for transformations that either: (i) produce goods in the set of requested goods; or (ii) consume goods from the available goods. After selecting the winning bids (the best set of transformations), the auctioneer updates the set of requested and available goods after the execution of these transformations. Moreover, the winning bids are included as part of the supply chain. Thereafter, the auctioneer starts a new auction in the sequence. The process continues until no bids can improve the supply chain. Hence, the purpose of the auctioneer is to use a sequence of auctions to progressively build the structure of the supply chain.

Figure 1 illustrates the operation of an SMA. Say that a cocktail bar intends to form a supply chain using an SMA to produce a gin & lemon cocktail. Assume that the bar knows approximate market prices for a gin & lemon cocktail as well as for its ingredients. The auctioneer starts the first auction in the SMA issuing a request for quotation (RFQ) for a gin & lemon cocktail. Figure 1a depicts the RFQ along with each good's market price in brackets (e.g. the expected market price of 1 liter of gin is 4€). During the first auction, the auctioneer received two bids: one offering to deliver a cocktail for 9€ (figure 1b); and another one to make a cocktail for 1€ when provided with lemon and gin (figure 1c). The auctioneer must now choose the winning bid out of the bids in figure 1d. However, notice that the bid in figure 1c can only be used whenever some provider(s) offer gin and lemon. Thus, the auctioneer assesses the *expected price* of the bid using

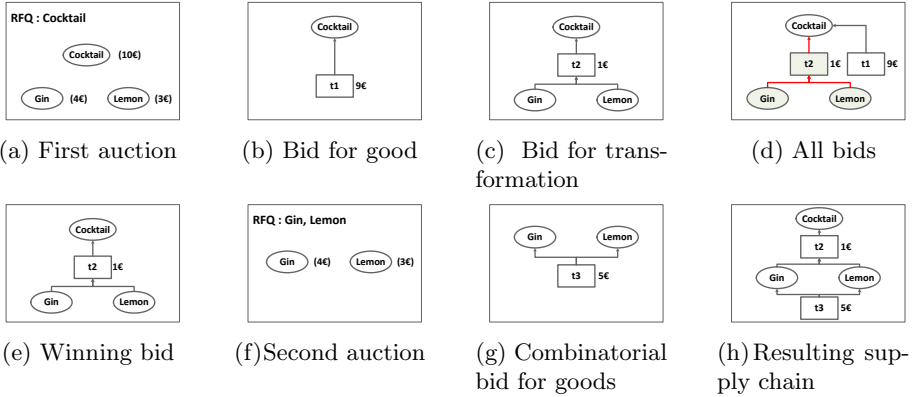


Fig. 1. Example of sequential mixed auction

the market prices of gin (4€) and lemon (3€). Since the expected price is $8(= 1 + 4 + 3)$ €, the auctioneer chooses this bid as the winning bid and discards the bid in figure 1b, namely buying the cocktail for 9€.

At this point, the structure of the supply chain is the one depicted in figure 1e. Nonetheless, in order to run the supply chain, the auctioneer must still find providers of gin and lemon. With this aim, the auctioneer starts a new auction of the SMA by issuing an RFQ for gin and lemon (figure 1f). According to our example, this time the auctioneer only receives the combinatorial bid in figure 1g which offers both lemon and gin for 5€. Since the bid is cheaper than the overall market price of both gin and lemon ($4€ + 3€$), this bid is selected as the winning bid of the second auction. Figure 1h shows the resulting structure of the supply chain after the second auction. Since there are no further goods to allocate, the auctioneer closes the SMA. The resulting supply chain produces a cocktail at the cost of 6€.

Although the SMA in this example obtains the optimal solution, this is not always the case. In general, at the end of each auction the auctioneer discards some bids because other bids are *expected* to lead to cheaper solutions. For instance, the bid in figure 1b is discarded to favour the bid in figure 1c. Therefore, since discarded bids might eventually lead to better solutions during subsequent auctions, unlike an MMUCA, an SMA is not guaranteed to obtain an optimal solution (sequence of transformations). Although SMAs may lose optimality, the example anticipates how an SMA help cope with computational complexity and bidders' uncertainty. Firstly, an SMA breaks the formation of a supply chain into several auctions, instead of running a single auction with all bids as MMUCA does. Secondly, after each auction in an SMA, bidders are informed about the needs of the supply chain. Therefore, the auctioneer guides bidders after each tier of the supply chain is formed, hence reducing their uncertainty with respect to participating in MMUCAs (MMUCA bidders only know the expected final outcome of the supply chain!).

3.1 Defining the Winner Determination Problem

An SMA is essentially a sequence of auctions (henceforth step auctions). For instance, the SMA in figure [1](#) is composed of two consecutive auctions. Each step auction receives a set of stock goods and final goods along with the bids submitted by bidders. Then the auctioneer solves the WDP to assess the winning bids as well as the remaining stock goods and required final goods, which are passed on to the next auction in the sequence. When solving the WDP, we assume that the auctioneer is aware of the market prices of goods so that it can compute the expected price of bids when necessary. The sequence of auctions continues till an auction either: (i) obtains a set of winning bids that produce the required goods while consuming all the stock goods; or (ii) does not receive any bids that can improve the supply chain. At this point, the winning bids of the last step auction stand for the SMA solution.

Next, we focus on formally defining the WDP faced by the auctioneer during each step auction of an SMA. Henceforth, we consider that the auctioneer holds a multi-set \mathcal{U}_{in} of stock goods to begin with and expects to end up with a multi-set \mathcal{U}_{out} of required (needed) goods. These are the input to the first auction in the SMA. For the formal definition of the WDP, we restrict ourselves to bids in the XOR-language, which is known to be fully expressive. Let C be the set of bidders. Let B be the set of all atomic bids. An atomic bid $b = (\mathcal{D}_b, p_b, \beta_b)$ consists of a multiset of transformations, a price, and a label indicating the owner of the bid, i.e. $\mathcal{D}_b \in \mathbb{N}^{(\mathbb{N}^G \times \mathbb{N}^G)}$, $p_b \in \mathbb{R}$, and $\beta_b \in C$. Let B_β be the set of all atomic bids submitted by bidder $\beta \in C$. Note that a bid can offer several units of the very same transformation. For each bid b , let t_{bk} be a unique label for the k th different transformation offered by bid b . Let $(\mathcal{I}_{bk}, \mathcal{O}_{bk})$ be the actual transformation labelled by t_{bk} .

At the l -th step auction in an SMA, let T_l be the set of labels t_{bk} for all transformations mentioned anywhere in the bids received by the auctioneer. The auctioneer has to decide which transformations to accept and the order to implement them. Thus, an allocation sequence Σ_l is an ordered list of a subset of the transformations in T_l . We write $t_{bk} \in \Sigma_l$ to say that the k -th transformation in bid b has been selected and $|\Sigma_l|_{t_{bk}}$ for the number of times that t_{bk} appears in Σ_l . Intuitively, an allocation sequence for a step auction is a valid solution iff: (i) it fulfills the semantics of the bidding language; (ii) it inherits all the transformations in the valid solution of the previous step auction while preserving their ordering; and (iii) the new transformations in the sequence (not inherited from the previous step auction) offer to either buy produced goods or sell required goods from the previous step auction. The last condition ensures that the new transformations accepted by the auctioneer either provide goods required to use the transformations in the previous step auction or consume goods produced in the same step auction. We are now ready to define under what circumstances a sequence of transformations constitutes a valid solution for a step auction in an SMA.

Definition 1 (Valid solution of a step auction). *Given an SMA, let Σ_{l-1} be a valid solution, and Buy_{l-1} and $Sell_{l-1}$ the multi-sets presenting the required*

and available goods after the $(l - 1)$ -th step auction. An allocation sequence Σ_l of the l -th step auction for a given set of atomic bids B is said to be a valid solution iff:

1. Σ_l either contains all or none of the transformations belonging to the same atomic bid.
2. Σ_l does not contain two transformations belonging to different atomic bids by the same bidder.
3. Each transformation in Σ_{l-1} belongs also to Σ_l .
4. Σ_l preserves the order of transformations in Σ_{l-1} . Thus, for every two transformation $t, t' \in \Sigma_{l-1}$, if t appears before t' in the allocation sequence Σ_{l-1} , it also appears before t' in the allocation sequence Σ_l .
5. For each transformation in Σ_l that is not in Σ_{l-1} , either some of its input goods are in $Sell_{l-1}$, some of its output goods are in Buy_{l-1} , or both.

For the first auction of the SMA, $\langle \{ \}, \mathcal{U}_{out}, \mathcal{U}_{in} \rangle$ stand for the valid solution, and the needed and stock goods.

In order to assess the expected revenue of a valid solution Σ_l , we must first compute the goods that the auctioneer should buy and sell in the market to implement the solution, namely to *use* all the transformations in the sequence. First, we compute the units of each good produced by a sequence Σ_l as:

$$P_l(g) = \sum_{t_{bk} \in \Sigma_l} |\Sigma_l|_{t_{bk}} \cdot [\mathcal{O}_{bk}(g) - \mathcal{I}_{bk}(g)] \quad (1)$$

Hence, we can obtain the number of units of each good held by the auctioneer after all the transformations in the sequence are used as:

$$Q_l(g) = \mathcal{U}_{in}(g) + P_l(g) - \mathcal{U}_{out}(g)$$

Now, we assess the units of each good to buy or sell in the market as:

$$Buy_l(g) = \begin{cases} 0 & \text{if } Q_l(g) > 0 \\ -Q_l(g) & \text{otherwise} \end{cases} \quad Sell_l(g) = \begin{cases} Q_l(g) & \text{if } Q_l(g) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Notice that in fact Buy_l and $Sell_l$ stand for the remaining required goods and available stock goods that step auction l passes on to the next auction. Now, assuming that the auctioneer knows the expected market prices at which goods can be bought ($P^- : G \rightarrow \mathbb{R}$) and sold ($P^+ : G \rightarrow \mathbb{R}$), the acutioneer can compute the *expected* revenue of a valid solution.

Definition 2 (Expected revenue of a valid solution). *The expected revenue of a valid solution Σ_l for step auction l is the sum of the prices associated with the selected atomic bids minus the expected prices of the goods that are required to be bought (Buy_l) plus the expected prices of the goods that are sold ($Sell_l$) in the market, namely:*

$$\sum_{b \in Selected(\Sigma_l)} p_b - \sum_{g \in G} Buy_l(g) P^-(g) + \sum_{g \in G} Sell_l(g) P^+(g)$$

where $Selected(\Sigma_l) = \{b \in B | \exists k : t_{bk} \in \Sigma_l\}$ is the set of selected atomic bids.

Definition 3 (WDP of a step auction). *Given multi-sets \mathcal{U}_{in} and \mathcal{U}_{out} of initial and final goods for an SMA, a set of atomic bids B for step auction l , a valid solution Σ_{l-1} of step auction $l-1$, and Buy_{l-1} and $Sell_{l-1}$ contain the units of each good to buy or sell in the market after step auction $l-1$, the winner determination problem for step auction l is the problem of finding a valid solution Σ_l that maximizes the expected revenue for the auctioneer.*

We say that a sequence of auctions is *complete* when the solution to the last step auction (n) in the sequence either: (i) produces all the required goods and consumes all the stock goods ($Buy_n = \{\}$ and $Sell_n = \{\}$); or (ii) is equal to the solution to the previous step auction (namely $\Sigma_n = \Sigma_{n-1}$). The first condition occurs when the auctioneer fully satisfies his initial requirements, while the second condition occurs when the auctioneer cannot find further bids that improve the current solution, hence the current supply chain. Whatever the case, the valid solution Σ_n contains the solution to the SMA.

4 Solving an SMA Step by Means of a Mixed Integer Linear Program

As outlined above, solving an SMA amounts to solve the WDP for the first step auction, then for the second step auction and so on and so forth until the SMA is complete. Hence, the key computational problem is how to solve the WDP for a step auction. In this section, we first summarise the Connected Component IP Solver (CCIP), which maps the original MMUCA WDP into a mixed integer program. Then, we modify it to solve an SMA step auction.

4.1 Connected Component IP Solver (CCIP)

In order to create a linear program we need to determine the maximum length of the allocation sequence. The worst case is assuming that every possible bid is accepted and hence every transformation is used. The total number of transformations (including multiple copies of the same transformation) can be assessed as $r = \sum_{b \in B} |\mathcal{D}_b|$.

Let T_b be the set of all t_{bk} for bid b . Let T be the set of all t_{bk} , namely the set of all distinguishable transformations (no copies of the very same transformation) mentioned anywhere in the bids. The auctioneer has to decide which transformations to accept and the order to implement them. At this aim, we represent each solution with a partial sequence $J : \{1, \dots, r\} \rightarrow T$. We employ the following decision variables: x_b is a binary variable that takes value 1 if bid b is accepted; x_{bk}^m is a binary variable that takes value 1 only if transformation t_{bk} is selected at the m -th position within the solution sequence (i.e. $J(m) = t_{bk}$).

Let S be a solution template, restricting the positions at which transformations can be executed. For each position m , $S(m)$ is a set containing the subset of transformations that can be executed at position m in the sequence. For more

information on how to define solution templates see [4]. We will only allow as solutions partial sequences fulfilling S . Hence we only have to create those variables x_{bk}^m such that $t_{bk} \in S(m)$.

We introduce two additional definitions to ease encoding the solver. First, x_{bk} represents the number of times that transformation t_{bk} is used in the solution sequence, assessed as:

$$x_{bk} = \sum_{m \in S^{-1}(t_{bk})} x_{bk}^m.$$

Second, we need to see how to assess the stock of a given good after m steps in terms of the decision variables we have defined. Say that we represent with the multiset of goods $Stock^m$ the quantity of resources available to the auctioneer after performing m steps.

Now, $\mathcal{P}^m(g)$, the units of good g available at the end of step m can be assessed as

$$\mathcal{P}^m(g) = \sum_{i=1}^m \sum_{t_{bk} \in S(m)} x_{bk}^i \cdot (\mathcal{O}_{bk}(g) - \mathcal{I}_{bk}(g)).$$

That is, for each time step from 1 to m , we add the output goods of the transformation executed at that step and remove its input goods. Hence, the stock of good g after step m is:

$$Stock^m(g) = \mathcal{U}_{in}(g) + \mathcal{P}^m(g) \quad (2)$$

Finally, the constraints that a valid solution has to fulfil in solver CCIP are:

1. We enforce that, whenever a bid is selected each of the transformations in that bid is used as many times as it is offered in the bid. Formally:

$$x_{bk} = x_b \cdot |\mathcal{D}_b|_{t_{bk}} \quad \forall b \in B, \quad \forall k \in \{1, \dots, |T_b|\} \quad (3)$$

where $|\mathcal{D}_b|_{t_{bk}}$ is the multiplicity of transformation t_{bk} in bid b .

2. We enforce that at most one bid can be accepted per bidder (XOR constraint):

$$\sum_{b \in B_\beta} x_b \leq 1 \quad \forall \beta \in C. \quad (4)$$

3. We enforce that enough goods are available to use the corresponding transformations at each position of the solution sequence. This constraint, represented by equation [5] below, is only needed for some of the positions (see [3] for a detailed description on L_F , the set of positions in which it must be enforced). Formally:

$$Stock^{m-1}(g) \geq \sum_{t_{bk} \in S(m)} x_{bk}^m \cdot \mathcal{I}_{bk}(g) \quad \forall g \in G, \quad \forall m \in L_F \quad (5)$$

4. We enforce that, after having performed all the selected transformations, the goods held by the auctioneer must be more than the goods that he requested, namely at least \mathcal{U}_{out} :

$$Stock^m(g) \geq \mathcal{U}_{out}(g) \quad \forall g \in G \quad (6)$$

Hence, solving the MMUCA WDP amounts to maximising the objective function:

$$\sum_{b \in B} x_b \cdot p_b \quad (7)$$

subject to inequations [3](#) to [6](#).

4.2 Solving the WDP for an SMA Step

Next we focus on modifying the CCIP solver so that it can solve an SMA step auction. Recall that $\Sigma_{l-1} = \langle t^1, \dots, t^q \rangle$ is the sequence describing the transformations accepted in a previous auction and their execution order and that B contains the set of bids received at this step.

There are two modifications that need to be put in place. First, we need to ensure that every transformation accepted in the previous SMA step, does also appear in solution of the current step and that the order in which they are executed is also maintained in the new solution. Second, we need to take into account the possibility of buying from and selling to the market.

To ensure that all the transformations accepted in the previous SMA step, do also appear in solution of the current step we add an additional bidder to C , and a single atomic bid to B , namely b_0 . b_0 is a combinatorial bid offering the set of all the transformations accepted in the previous SMA step (that is, the transformations in Σ_{l-1}) at no cost. Hence, from now on we can refer to any transformation $t^k \in \Sigma_{l-1}$ as t_{b_0k} . Now, to ensure that this bid is taken, we add the additional constraint

$$x_{b_0} = 1. \quad (8)$$

We do also need to ensure that the ordering in the sequence of transformations accepted in the previous SMA step is maintained in the solution of this step. To encode this, for any transformation t_{bk} we define the time at which it is fired as $f_{bk} = \sum_{m \in S^{-1}(t_{bk})} m \cdot x_{bk}^m$. Now, for each consecutive pair of transformations we need to ensure that they are placed in the correct order in the solution sequence of this SMA step. That is:

$$f_{b_0k} \leq f_{b_0k+1} \quad \forall k \in \{1, \dots, q-1\}. \quad (9)$$

We take into account the possibility of buying from and selling to the market by introducing a new integer decision variable y_g , for each good at trade. y_g represents the number of units of good g that will be bought directly from the market. We can assume that goods are bought from the market before the solution sequence starts its execution. Hence, we can modify the expression for $Stock^m$ in equation [2](#) to consider the goods bought from the market:

$$Stock^m(g) = \mathcal{U}_{in}(g) + \mathcal{P}^m(g) + y_g. \quad (10)$$

Furthermore, we need to modify the objective function in [7](#) so that it takes into account the costs of buying from the market and the benefits obtained by selling to the market as described in definition [2](#). Note that $Sell(g)$, the units of good g that will be sold to the market can be assessed as the stock for that good after executing the last step minus the requirements of the auctioneer for that good, that is $Sell(g) = Stock^r(g) - U_{out}(g)$. The objective function for a step auction is written as:

$$\sum_{b \in B} x_b \cdot p_b - \sum_{g \in G} y_g \cdot P^-(g) + \sum_{g \in G} Sell(g) \cdot P^+(g) \quad (11)$$

Hence, solving the SMA step WDP amounts to maximising objective function [11](#) subject to constraints [8](#) and [9](#) and to the regular MMUCA constraints [3](#) to [6](#) with the definition of $Stock$ modified as in equation [10](#).

5 Conclusions and Future Work

In this work, we continue the approach introduced in [5](#) to make mixed auctions applicable to supply chain formation in real-world procurement scenarios.

Following [5](#), to cope with the extensive computing times of MMUCA and bidder's uncertainties we moved the supply chain formation process from a single auction to a sequence of auctions. At each step auction, bidders are only allowed to bid on transformations that consume available goods or produce requested goods. After selecting the best set of transformations, the auctioneer updates the set of requested and available goods. The sequence ends when supply chain cannot be further improved. Each auction deals with just a small part of the supply chain. Thus, while solving the WDP for an individual auction we deal with small subsets of bidders, goods and transformations of former MMUCA. Preliminary results in [5](#) have shown savings on solution times up to 6 times while maintaining a reasonable quality.

The main contribution of this paper, the mapping of a step auction to a mixed integer linear program, is less restrictive than the approach introduced in [5](#), based on keeping a strict ordering among transformations. Hence, future experimental results using this mapping are expected to increase solution quality.

Acknowledgements. Work funded by projects EVE (TIN2009-14702-C02-01 and 02), AT (CONSOLIDER CSD2007-0022), Generalitat de Catalunya (2009-SGR-1434) and CSIC 201050I008.

References

1. Cerquides, J., Endriss, U., Giovannucci, A., Rodriguez-Aguilar, J.A.: Bidding languages and winner determination for mixed multi-unit combinatorial auctions. In: IJCAI, Hyderabad, India, pp. 1221–1226 (2007)

2. Cramton, P., Shoham, Y., Steinberg, R. (eds.): *Combinatorial Auctions*. MIT Press (2006)
3. Giovannucci, A.: *Computationally Manageable Combinatorial Auctions for Supply Chain Automation*. PhD thesis, Universitat Autònoma de Barcelona (2007)
4. Giovannucci, A., Vinyals, M., Cerquides, J., Rodríguez-Aguilar, J.A.: *Computationally-efficient winner determination for mixed multi-unit combinatorial auctions*. In: *AAMAS*, Estoril, Portugal, May 12-16, pp. 1071–1078 (2008)
5. Mikhaylov, B., Cerquides, J., Rodríguez-Aguilar, J.A.: *Sequential mixed auctions for supply chain formation*. In: *International Conference on Electronic Commerce* (2011)
6. Nisan, N.: *Bidding Languages for Combinatorial Auctions*. ch. 9 *Combinatorial Auctions*. MIT Press (2006)
7. Walsh, W.E., Wellman, M.P.: *Decentralized supply chain formation: A market protocol and competitive equilibrium analysis*. *Journal of Artificial Intelligence Research* 19, 513–567 (2003)

Global Feature Subset Selection on High-Dimensional Datasets Using Re-ranking-based EDAs

Pablo Bermejo, Luis de La Ossa, and Jose M. Puerta

Edificio I³A, Albacete, Castilla-La Mancha University, Spain
{Pablo.Bermejo,Luis.Delaossa,Jose.Puerta}@uclm.es

Abstract. The relatively recent appearance of high-dimensional databases has made traditional search algorithms too expensive in terms of time and memory resources. Thus, several modifications or enhancements to local search algorithms can be found in the literature to deal with this problem. However, non-deterministic global search, which is expected to perform better than local, still lacks appropriate adaptations or new developments for high-dimensional databases. We present a new non-deterministic iterative method which performs a global search and can easily handle datasets with high cardinality and, furthermore, it outperforms a wide variety of local search algorithms.

1 Introduction

Lately many new domains of expertise have arisen which use or produce loads of information greater than another traditional fields of study. Thus, while some problems traditionally being described by tens or hundreds of variables can easily benefit from machine learning techniques, now new problems such as face recognition, text data mining or microarray expressions need thousands or tens of thousands variables to describe one single instance. While low-dimensional problems could easily benefit from Feature Subset Selection (FSS) [13] methods in order to find the most relevant variables, high-dimensional databases make traditional FSS methods suffer the consequences of lack of resources for computation and unfeasible run-time complexity. This way, there is a need to adapt new FSS methods or create new ones to deal with the high-dimensional databases in order to reduce the number of variables so that they can benefit from the following CLUB advantages:

1. **Compactness.** Producing a more compact database without losing the semantic meaning of the variables, as it happens with another reduction techniques like Principal Component Analysis (PCA) [15].
2. **Lightness.** Besides, these models are built needing fewer computational resources.
3. **Understandability.** Predictive models built after the reduced database are more easily understood by the domain experts than models built from thousands of variables.
4. **Better.** Models built are theoretically free from redundant or irrelevant variables so they are expected to perform better.

Several taxonomies can be found in the literature in order to classify *supervised* FSS methods. If we attend to the metric computed to score each subset of attributes, we can find *filter*, *wrapper* and *hybrid* methods. Filter methods make a fast computation of mathematical or intrinsic properties of data, commonly with respect to the class; while wrapper methods build a predictive model and use its goodness as score. The wrapper approach is much slower than the filter one, but it usually gets better results because directly evaluate the candidate subset by using the target classifier, on the other hand filter methods are faster and un-biased to the classifier to be used later, but usually they approximate the goodness of a subset by using marginal statistics. In the literature, we can also find hybrid approaches, which combine filter and wrapper search to benefit from the singular advantages of each methodology.

Leaving aside exhaustive search and attending to the direction followed in the search space, we can distinguish between *local* and *global* search. Local search algorithms, such as sequential [16], start search from a given point and explore the space of solutions in a neighborhood-based manner, having a great likelihood of getting stuck in a local optima, which sometimes is alleviated adding some randomness to the process as in GRASP [9]. On the other hand, global search algorithms, such as Genetic Algorithms (GAs) [22] or Estimation of Distribution Algorithms (EDAs) [17], perform the search in an stochastic manner in order to explore a larger portion of the search space, thus trying to approximate to the global optima. Global algorithms usually obtain better results than local, but the recent appearance of high-dimensional databases have made them unfeasible in many cases. Thus the main contribution in this work is to propose and evaluate the design of a new non-deterministic FSS global search algorithm which performs a smart hybrid search based on blocks portions of the features. Concretely an EDA is used as search engine and executed several times over the proposed blocks. From the experiments we can observe that our proposal outperforms FSS state-of-the-art algorithms for high-dimensional datasets, when studying the balance between the three most common evaluation criteria: classification accuracy, cardinality of the selected subset and number of evaluated subsets (cpu-time requirements).

Next section introduces several state-of-the-art search algorithms found in the literature, including some adaptations to deal with datasets with high cardinality. Section 3 presents our proposal for non-deterministic global search over high-dimensional databases; then in Section 4 we describe a detailed evaluation of our proposal and comparisons with well-known search algorithms. Finally in Section 5 we summarize the main conclusions obtained.

2 Search Algorithms Applied to High-Dimensional Databases

Several traditional local search algorithms, such as Hill Climbing (HC) or Sequential Feature Selection (SFS), which explore the space of solutions in a greedy manner, and global search algorithms such as GAs and EDAs, are very expensive (or unfeasible) in terms of wrapper evaluations as the size of databases increases. That is why we can find in the literature some alleviations in the design of the search in order to continue benefiting from these algorithms.

2.1 Local Search Algorithms

SFS starts with an initial subset of selected features (commonly $S = \emptyset$), and performs a forward greedy selection with (usually) wrapper evaluation at each step. When no addition of any feature to S improves the current best performance, then the stop criterion is triggered and the search is finished. Worst case complexity of *SFS* is quadratic $\mathcal{O}(n^2)$ which makes it a too expensive algorithm when dealing with datasets having tens of thousands of attributes.

Recently, we can find in the literature algorithms like *BIRS* [20] or *IWSS* [2] which perform an alleviation of pure wrapper algorithms in the way of a filter-wrapper (hybrid) search, commonly using a ranking computed after an information-based measure like Information Gain of Symmetrical Uncertainty as in [11]. Thus, a former uni-variaded filter ranking is done for all predictive features respect to the class, and then a sequential wrapper is carried out over the ranking in a best-first manner. Algorithms following this approach like *BIRS* and *IWSS* have linear complexity, i.e. they carry out exactly n wrapper evaluations.

When performing a sequential search over a filter uni-variaded ranking, since (in)dependencies among variables are not caught by the filter measure used to create the ranking, we may find two problems:

1. During the search, some of the currently selected attribute might become irrelevant after adding new ones and it would be desirable to get rid of them.
2. Some attributes placed in the end of the (marginal) ranking might become more relevant than former attributes once one or more attributes have been selected during the search, so the search could be improved if we use a dynamic ranking instead of a static one.

A solution to problem (1) is found in [3] with the *IWSS_r* algorithm, which is based on *IWSS* but with the add-on in its wrapper search of not only testing the addition of a new attribute, but to swap it by any one of the already selected attributes; however this improvement increases again the worst-case complexity to $\mathcal{O}(n^2)$ as *SFS*, although its in-practice complexity is by far smaller. Algorithm *BARS* [19] is another solution to this problem which splits the ranking in two and tries to merge attribute subsets from both splits, having a worse theoretical complexity than *IWSS_r* but being quite efficient in practice.

Problem (1) can be seen as a consequence of problem (2), which is tackled in [1] by using a *re-ranking* approach. Re-ranking is a methodological improvement to sequential search algorithms, and thus it can be applied to the aforementioned algorithms *SFS*, *IWSS*, *IWSS_r* and *BARS*.

Starting with a set of selected features $S = \emptyset$, the application of re-ranking to a sequential search algorithm A consists on: (1) compute ranking R given $I(X_i; C|S)$ for each predictive feature X_i respect to the class C , (2) run A over first B features on R and insert selected features in S ; (3) do $R \leftarrow R - S$; (4) if S has been increased in step (2), return to (1), stop search otherwise.

Thus, the idea behind re-ranking is to dynamically adapt the ranking by moving to former positions attributes first ranked because a low marginal score $I(X_i; C|\emptyset)$ but having a higher (conditional) score once new attributes have been selected and added

to S . The problem here is that as more attributes are selected and S increases, the computation of $I(X_i; C|S)$ becomes soon intractable so approximate methods must be used. In [11] we can find a comparison of 3 approximations to compute $I(X_i; C|S)$ in a FSS setting, finding that the best method for re-ranking is CMIM [10]:

$$I(X_i, C|S) \approx \min_{X_j \in S} I(X_i; C|X_j)$$

The results of improving SFS, IWSS, $IWSS_r$ and BARS with re-ranking is an astonishing reduction in the number of wrapper evaluations (sub-linear complexity) during the sequential search, while maintaining the same accuracy level. Thus, sequential algorithms greatly benefit from this approach and can easily support search over high-dimensional databases.

Besides sequential algorithms, we can find meta-heuristics which improve local search by adding some randomness to the process, such as GRASP. This meta-heuristic obtains several solutions by using a randomized constructive algorithm, which are later optimized by using local search. Casado [6] presents a GRASP algorithm for feature selection which needs a pre-fixed number of features and there is not any proposed automatic method to decide this number, what reduces the flexibility of the algorithm. In [8] a proposal is made which makes use of hybrid evaluations, but the use of standard techniques in the local search phase of GRASP makes the resulting algorithm prohibitive for its use in high-dimensional. It is in [4] where we first find the proposal of a GRASP algorithm which is designed for application on high-dimensional databases, combining a stochastic IWSS search with Hill Climbing, resulting in an algorithm with sub-linear complexity and with very competitive final subsets selected; in the following we refer to this algorithm as *GRASP_{hc}*.

2.2 Non-deterministic Global Search Algorithms

Stochastic global search algorithms are expected to catch the underlying interdependencies among all variables. That is why evolutionary algorithms such as GAs and EDAs usually obtain better results than local search ones. However, the search space defined when facing FSS problems by this type of algorithms has cardinality 2^n , which makes them unfeasible when dealing with high-dimensional datasets. In order to benefit from the advantages of these global search algorithms when dealing with FSS in high-dimensional datasets, several proposals based on GA and EDA can be found in the literature.

The first EDA-based proposal for FSS can be found in [14], in which authors codify probability distributions using bayesian networks; however, this is a solution unfeasible for high-dimensional datasets, which is the aim of this work. In [5], authors use EDAs for feature selection in two datasets (2000 and 7129 features) and they focus on deciding how to set the variables' probabilities when initializing the search, since they state that this is the bottleneck in EDAs when dealing with thousands of features. They propose three initializations methods based on the subset selected by a previous SFS search over all the dataset, and thus the expected number of set variables in each individual of the first generation is the cardinality of the reduced subset selected by SFS. Although those three methods perform better than using an uniform distribution and alleviate the

high-dimensionality problem for EDAs, they require a whole SFS search a-priori, what leads us to (worst-case) $\mathcal{O}(n^2)$ complexity just at initialization time. Another simplification is found in [23], where they propose a memetic algorithm which is evaluated with low-dimensional databases and also for microarrays having over 1000 features, in which case they simplify the problem by setting a top threshold of 50 set features in each individual. This way the global search is truncated in a non-informed way and interdependencies among variables are lost. Other prior simplification on the number of features to use as input to the search algorithm is in [21], where authors perform feature subset selection using a GA, and the input variables are first filtered performing t-test and ANOVA tests to compute the significance of each variable respect to the class to predict. The problem we identify here is again that the performance of the global search might be downgraded since features judged as non-relevant, could be indeed relevant given another set of features.

3 Re-ranking-Based EDA

As shown in the previous section, although several improvements have been developed for successfully running local search algorithms over high-dimensional datasets, there is still a need of solutions for non-deterministic global search which can fulfill these 2 constraints: (1) need of sub-quadratic complexity and (2) avoid the a-priori marginal-based selection of a subset of attributes. Up to the authors' knowledge, our proposal is the first one to fulfill these requirements: Re-ranking-based EDA (FSS-rEDA).

Re-ranking (see Section 2.1) has proved to improve wrapper and hybrid sequential algorithms in terms of final subset cardinality and to dramatically decrease the number of wrapper evaluations. Thus, since global search algorithms are expected to perform better than local, the idea behind FSS-rEDA is to adapt the re-ranking approach to EDAs with the goal of performing better than the mentioned local search algorithms. Besides, the FSS-rEDA proposal must fulfill the two aforementioned constraints. Figure 1 shows the structure of the FSS-rEDA algorithm proposed in this work.

The FSS-rEDA procedure consists on iteratively applying an EDA search over the first block (block size depends of the stage of the algorithm) of a dynamic ranking of attributes to get subset \mathcal{S} , which is used to re-rank the remaining attributes and appended at the beginning of the new ranking. Thus, all attributes have a chance of being studied by the global search algorithm (EDA), because their position in the ranking can change from an iteration to the next one depending in the current content of \mathcal{S} . Lines 3-5 in Figure 1 initiate ranking R for the first time, inserting features X_i in decreasing order given the uni-variate filter metric I of X_i respect to the *class*. Lines 7-8 in the loop cut the first B' features from ranking R and create a temporal ranking R' with the cut features. Next, line 9 runs an EDA search and returns a subset of attributes (which will be used in line 13 to re-rank R). Line 10 uses a \triangleright comparison as in IWSS and IWSS_r algorithms (Section 2.1); concretely, a 5-fold cross (5cv) validation is computed and then, a subset performs better than another if not only the mean accuracy is greater, but also the accuracy in at least 2 out of the 5 folds.

Line 12 is used for the cases in which subset \mathcal{S} is too large compared to current block size, so block size is increased for next global search step. Line 13 re-ranks the

In B block size, \mathbf{T} training set, \mathcal{C} classifier, $\mathcal{S}_{best} = \emptyset$
 Out \mathcal{S} final selected subset

```

1  $R = \{\}$  // first ranking
2  $B' = B$ 
3 for each attribute  $X_i \in \mathbf{T}$ 
4    $Score = I(\mathbf{T}, X_i, \text{class})$ 
5   insert  $X_i$  in  $R$  according to  $Score$ 
6 while(true)
7    $R' \leftarrow$  first  $B'$  attributes in  $R$ 
8    $R \leftarrow R - R'$ 
9    $\mathcal{S} \leftarrow \text{EDA}(R', \mathbf{T}, \mathcal{C})$ 
10  if(evaluate( $\mathcal{C}, \mathcal{S}_{best}, \mathbf{T}$ )  $\triangleright$  evaluate( $\mathcal{C}, \mathcal{S}, \mathbf{T}$ )) return  $\mathcal{S}_{best}$ 
11   $\mathcal{S}_{best} \leftarrow \mathcal{S}$ 
12  if( $B' < 2 \times |\mathcal{S}|$ )  $B' = 2 \times |\mathcal{S}|$ 
13   $R \leftarrow \text{rerank}(R|\mathcal{S})$ 
14   $R \leftarrow \text{concatenate}(\mathcal{S}, R)$ 

```

Fig. 1. Re-ranking-based EDA for feature subset selection (FSS-rEDA)

attributes computing the same information score as in line 4, but conditioning it to current subset \mathcal{S} ; in order to do this, we use the CMIM approximation mentioned in Section 2.1. The actual EDA-based search is performed in line 9. In this work we used the Univariate Model Distribution Algorithm (UMDA) [18] which assumes marginal independence between variables when learning their joint probability. We define an individual as a bit set where bit i makes reference to feature $R'[i]$. If a bit is selected to 1 then the feature is selected in that individual, and the contrary happens if it is set to 0. Notice that at each step, the search space for EDA application is 2^r , r being the block size ($r \ll n$). The initial population is initiated giving to each bit in the individual a probability proportional to the score computed for that attribute when creating the ranking. The fitness function used to measure the merit of each individual is:

fitness = $\alpha * \text{accuracy} + (1 - \alpha) * \text{cardinalityReduction}$
 $\alpha \in [0, 1]$ and $\text{cardinalityReduction} = 1 - (|\mathcal{S}| / |\text{originalDataset}|)$.

Thus, greater values of α give more priority to accuracy over cardinality reduction of the selected subset. Besides, the UMDA need the specification of several parameters (population size, number of generations and convergence criterion). Some of them are tuned in the section 4.

4 Experiments

4.1 Evaluation Corpus

We use a corpus with 11 high-dimensionality datasets¹ used in different fields of study: face recognition (warpAR10P, warpAR10P, pixraw10P and orlraws10P); genes expression microarrays (TOX-171, SMK-CAN-187, GLI-85, GLA-BRA-180, CLL-SUB-111

¹ Downloaded from the *ASU Feature Selection Repository*
<http://featureselection.asu.edu>

and CLL-SUB-111); and text mining (pcmac and basehock). These datasets are free from any imbalance problem and their cardinality ranges from 2400 to 46151 features.

4.2 Evaluation Methodology

Our goal is to compare FSS-rEDA to sequential algorithms: linear (IWSS), quadratic (SFS), and over quadratic (BARS) worst-case complexity, their adaptations to high-dimensional databases by using the re-ranking approach (SFS^R , $IWSS^R$, $BARS^R$) [1], and the non-deterministic local search GRASPhc.

First, we run FSS-rEDA over the evaluation corpus, using a Naive Bayes classifier (NB) and a 10 folds cross validation (apply search and build classifier from each of the 10 training folds, and evaluate on the corresponding 10 test folds). Besides, we tune the following parameters: α , *population size* and *maximum number of generations* (stop criterion); B is fixed to 20 since several experiments showed that increasing it does not improve results. Since FSS-rEDA and GRASPhc are non-deterministic algorithms, the reported results are the average over 10 independent runs. Second, we select the best configuration found for FSS-rEDA and we compare it to SFS, IWSS, BARS, SFS^R , $IWSS^R$, $BARS^R$ and GRASPhc, with the same corpus, classifier and validation procedure (10cv). In all the cases, a statistical analysis is used for comparison, using a Friedman test followed by a post-hoc Holm test, as suggested in [7] and using the code provided in [12]; the confidence level is set to 0.05. The control algorithm is automatically selected by the code used, commonly being the algorithm with the highest accuracy when comparing by accuracy, or the lowest cardinality when comparing by number of attributes. The comparison taking into account the three criteria is carried out as follows: first we compare by accuracy and cross-out (remove) those algorithms worse than the control algorithm; then we compare the remaining ones by cardinality of the selected subset and finally by using the number of wrappers evaluations. When in any state of the comparison only 4 or less algorithms remain to be compared, then a paired one-tail Wilcoxon signed-ranks test is used.

4.3 Results

FSS-rEDA Configuration. Table 1 shows results for FSS-rEDA considering three different values of the fitness factor α (0.7, 0.8, 0.9). With respect to *number of generations* and *population size*, they are tuned with values: 20, 30, 40 and 50, having always the same value for both parameters. Although it is common to find in the literature configurations for evolving algorithms with *number of generations* greater than *population size*, in our experience with FSS-rEDA, not only it is not worth but we get an increased number in evaluations. Our stopping criterion is to reach *number of generations*.

In Table 2 we show the statistical comparisons following the methodology indicated in Section 4.2. Each configuration of FSS-rEDA is represented by its average values shown in Table 1, and the column name is made 'P-G=' representing the Population size and number of Generations; and the α fitness value. As a conclusion, the best configuration found for our FSS-rEDA proposal is: population size and number of generations =40, and $\alpha = 0.9$. Thus, is the configuration used for comparison with the other algorithms.

Table 1. FSS-rREDA algorithm with UMDA, NB classifier with 10-CV and $B = 20$

Dataset	$\alpha = 0.5$			$\alpha = 0.7$			$\alpha = 0.9$		
	Acc.	#Atts	#Evals.	Acc.	#Att.	#Evals.	Acc.	#Att.	#Evals.
warpPIE10P	75.5	4.6	427.4	81.6	6.7	557.1	85.7	10.1	607.0
warpAR10P	57.7	3.5	349.8	65.8	5.5	467.3	67.3	8.8	541.2
pixraw10P	91.1	6.9	369.5	92.8	3.2	438.2	93.4	4.7	429.7
orlraws10P	83.6	3.7	528.8	86.7	4.4	470.1	89.4	6.7	494.0
TOX-171	59.5	3.0	298.1	68.3	6.3	540.3	71.8	12.0	767.6
SMK-CAN-187	61.8	1.5	127.3	68.5	3.7	346.7	68.0	7.4	436.2
GLI-85	82.0	1.9	255.1	84.9	3.2	342.6	89.4	6.4	344.7
GLA-BRA-180	65.4	1.9	223.3	66.7	3.1	389.8	66.7	7.4	540.0
CLL-SUB-111	70.1	2.5	306.2	73.7	4.7	487.3	77.0	8.4	527.5
pcmac	65.4	1.0	47.4	74.4	4.8	331.0	78.2	9.0	324.5
basehock	69.2	1.8	104.1	77.6	5.0	343.0	87.7	22.8	1349.5
Mean	71.0	2.9	276.1	76.4	4.6	428.5	79.5	9.4	578.4
Population size and #generations = 20									
warpPIE10P	76.4	4.7	666.3	81.8	6.5	854.0	84.1	9.3	957.6
warpAR10P	60.3	3.3	542.1	64.7	5.3	726.8	66.8	8.5	853.5
pixraw10P	90.3	4.8	643.6	91.0	3.0	646.0	94.3	4.4	690.5
orlraws10P	84.1	3.3	847.0	88.9	4.4	751.7	90.7	6.2	810.4
TOX-171	58.8	3.0	447.1	67.2	6.0	893.7	72.4	11.8	1348.2
SMK-CAN-187	62.8	1.5	189.0	68.4	3.3	538.2	69.4	7.2	774.7
GLI-85	81.4	1.6	379.0	84.7	3.1	536.2	89.3	6.0	545.9
GLA-BRA-180	66.4	1.7	326.7	66.3	3.1	593.2	67.0	6.9	832.6
CLL-SUB-111	67.8	2.5	457.2	72.0	4.3	749.2	77.0	8.2	822.6
pcmac	65.4	1.0	64.5	74.3	4.6	482.5	78.0	8.5	485.7
basehock	70.1	2.0	147.5	77.3	4.8	502.0	86.8	19.4	2157.2
Mean	71.3	2.7	428.2	76.1	4.4	661.2	79.6	8.8	934.4
Population size and #generations = 30									
warpPIE10P	76.0	4.4	878.9	81.3	6.4	1090.1	85.6	9.4	1294.2
warpAR10P	59.8	3.4	715.2	64.2	5.3	1035.1	68.9	8.4	1178.4
pixraw10P	90.3	3.4	982.1	91.6	2.7	871.5	94.4	4.3	929.3
orlraws10P	84.7	3.2	1142.0	87.9	4.4	979.8	91.0	5.8	1075.6
TOX-171	59.6	2.9	579.8	66.7	5.9	1183.6	72.0	11.7	1838.8
SMK-CAN-187	63.3	1.6	234.6	68.6	3.3	736.7	67.6	7.2	1014.3
GLI-85	82.0	1.8	510.9	85.2	3.1	755.3	89.0	5.8	748.2
GLA-BRA-180	65.2	1.8	417.6	65.4	3.0	854.3	66.4	6.9	1173.9
CLL-SUB-111	67.3	2.4	591.7	73.5	4.2	981.2	76.1	8.0	1131.5
pcmac	65.4	1.0	81.7	74.2	4.5	594.1	78.1	8.5	642.5
basehock	69.8	1.9	189.4	77.5	4.8	657.5	86.4	18.0	2900.4
Mean	71.2	2.5	574.9	76.0	4.3	885.4	79.6	8.5	1266.1
Population size and #generations = 40									
warpPIE10P	74.0	4.6	1130.6	81.1	6.3	1424.1	85.7	9.3	1653.9
warpAR10P	59.3	3.4	951.1	65.1	5.2	1248.2	67.4	8.3	1469.1
pixraw10P	91.7	2.7	1255.8	92.0	2.7	1082.0	95.0	4.4	1150.8
orlraws10P	81.9	3.0	1390.2	88.6	4.3	1230.5	89.9	5.8	1387.8
TOX-171	58.1	2.9	707.5	67.7	6.0	1476.9	71.2	11.5	2382.8
SMK-CAN-187	63.7	1.6	290.2	68.5	3.2	905.8	67.7	7.2	1285.9
GLI-85	82.1	1.7	625.2	84.4	3.0	882.8	88.5	5.5	931.5
GLA-BRA-180	65.2	1.8	544.7	64.7	2.9	958.1	67.1	7.0	1455.2
CLL-SUB-111	67.1	2.4	724.8	74.1	4.0	1254.6	75.9	7.9	1388.2
pcmac	65.4	1.0	97.3	74.1	4.4	724.3	78.0	8.2	760.2
basehock	70.0	2.0	226.8	77.4	4.7	803.3	85.8	16.0	3128.3
Mean	70.8	2.4	722.2	76.2	4.2	1090.1	79.3	8.3	1544.9
Population size and #generations = 50									

Table 2. Statistical Comparisons for Different Configurations of FSS-rEDA

	P-G=20			P-G=30			P-G=40			P-G=50		
	$\alpha=0.5$	$\alpha=0.7$	$\alpha=0.9$	$\alpha=0.5$	$\alpha=0.7$	$\alpha=0.9$	$\alpha=0.5$	$\alpha=0.7$	$\alpha=0.9$	$\alpha=0.5$	$\alpha=0.7$	$\alpha=0.9$
Acc.	71.0	76.4	79.5	71.3	76.1	79.6	71.2	76.0	79.6	70.8	76.2	79.3
#Atts.	2.9	4.6	9.4	2.7	4.4	8.8	2.5	4.3	8.5	2.4	4.2	8.3
#Evals.	276.1	428.5	578.4	428.2	661.2	934.4	574.9	885.4	1266.1	722.2	1090.1	1544.9

FSS-rEDA vs. Local Search Algorithms. Table 3 shows the statistical analysis for the comparison between all the algorithms considered in our experiments (detailed tables are not included because of lack of space) and the best configuration for our proposal FSS-rEDA. In the case of search algorithms with re-ranking, block size is also set to 20. For GRASPhc parameters, we set *multi-starts* = 50, ranking size=100, improving phase=HillClimbing, as suggested by authors in [4].

Table 3 also summarizes the results of the statistical analysis, and shows that FSS-rEDA is the best choice when taking into account the three criteria (accuracy, subset cardinality and number of required evaluations), therefore it stands as a very efficient global search algorithm to be applied over high-dimensional databases.

Table 3. Statistical comparisons for Local Search Algorithms and FSS-rEDA

	IWSS	SFS	BARS	IWSS ^R	SFS ^R	BARS ^R	GRASPhc	rEDA
Acc.	79.3	79.0	78.1	79.3	80.7	79.1	79.6	79.6
#Atts	23.1	23.8	10.7	14.2	22.9	12.6	13.1	8.5
#Evals.	12890.0	167162.2	38304.7	89.9	569.6	1389.4	8171.8	1266.1

5 Conclusions

We have proposed a new method which makes it possible to run a stochastic global search-based FSS algorithm over high-dimensional databases. After extensive statistical comparisons we show that our proposal outperforms in terms of cardinality of the selected subset to all local search algorithms and their enhancements compared. Furthermore, our proposal presents an in-practice sublinear complexity respect to the number of wrapper evaluations performed.

Acknowledgements. This work has been partially funded by FEDER funds, the Spanish Government (MICINN) through projects TIN2010-20900-C04-03 and PCI08-0048-8577.

References

- Bermejo, P., de la Ossa, L., Gámez, J.A., Puerta, J.M.: Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking, Knowledge-Based Systems (in press)
- Bermejo, P., Gámez, J., Puerta, J.: On incremental wrapper-based attribute selection: experimental analysis of the relevance criteria. In: IPMU 2008: Proceedings of the 12th Intl. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (2008)

3. Bermejo, P., Gámez, J.A., Puerta, J.M.: Incremental wrapper-based subset selection with replacement: An advantageous alternative to sequential forward selection. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009 (2009)
4. Bermejo, P., Gámez, J.A., Puerta, J.M.: A grasp algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. *Pattern Recognition Letters* 32(5), 701–711 (2011)
5. Blanco, R., Naga, P.L., Iñaki Inza, I., Sierra, B.: Selection of highly accurate genes for cancer classification by estimation of distribution algorithms. In: Workshop of Bayesian Models in Medicine, AIME 2001 (2001)
6. Casado-Yusta, S.: Different metaheuristic strategies to solve the feature selection problem. *Pattern Recognition Letters* 30(5), 525–534 (2009)
7. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
8. Essegir, M.A.: Effective wrapper-filter hybridization through grasp schemata. In: MLR Workshop and Conference Proceedings, Feature Selection in Data Mining, vol. 10 (2010)
9. Feo, T.A., Resende, M.G.: Greedy randomized adaptive search procedures. *Global Optimization* 6(2), 109–133 (1995)
10. Fleuret, F.: Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research* 5, 1531–1555 (2004)
11. Flores, J., Gámez, J.A., Mateo, J.L.: Mining the esrom: A study of breeding value classification in manchego sheep by means of attribute selection and construction. *Computers and Electronics in Agriculture* 60(2), 167–177 (2008)
12. Garcia, S., Herrera, F.: An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research* 9, 2677–2694 (2008)
13. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
14. Inza, I., Larrañaga, P., Etxebarria, R., Sierra, B.: Feature subset selection by bayesian network-based optimization. *Artificial Intelligence* 123, 157–184 (2000)
15. Jolliffe, I.: *Principal Component Analysis*. Springer, Heidelberg (1986)
16. Kittler, J.: Feature set search algorithms. *Pattern Recognition and Signal Processing*, 41–60 (1978)
17. Larrañaga, P., Lozano, J.A.: *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers (2001)
18. Mühlenbein, H.: The equation for response to selection and its use for prediction. *Evolutionary Computation* 5, 303–346 (1998)
19. Ruiz, R., Aguilar, J.S., Riquelme, J.: Best agglomerative ranked subset for feature selection. In: *JMLR: Workshop and Conference Proceedings*, vol. 4 (New Challenges for feature selection) (2009)
20. Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S.: Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recogn.* 39, 2383–2392 (2006)
21. Tan, Q., Thomassen, M., Jochumsen, K.M., Zhao, J.H., Christensen, K., Kruse, T.A.: Evolutionary algorithm for feature subset selection in predicting tumor outcomes using microarray data. In: Mändoiu, I., Wang, S.-L., Zelikovsky, A. (eds.) *ISBRA 2008. LNCS (LNBI)*, vol. 4983, pp. 426–433. Springer, Heidelberg (2008)
22. Yang, J., Honavar, V.: Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems* 13(2), 44–49 (1998)
23. Zhu, Z., Ong, Y.-S., Dash, M.: Wrapper-filter feature selection algorithm using a memetic framework. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 37(1), 70–76 (2007)

A Comparison of Two Strategies for Scaling Up Instance Selection in Huge Datasets*

Aida de Haro-García, Javier Pérez-Rodríguez, and Nicolás García-Pedrajas

Department of Computing and Numerical Analysis, University of Córdoba, Spain
{adeharo,npedrajas}@uco.es, javier@cibrg.org
<http://www.cibrg.org/>

Abstract. Instance selection is becoming more and more relevant due to the huge amount of data that is constantly being produced. However, although current algorithms are useful for fairly large datasets, many scaling problems are found when the number of instances is of hundred of thousands or millions. Most instance selection algorithms are of complexity at least $O(n^2)$, n being the number of instances. When we face huge problems, the scalability becomes an issue, and most of the algorithms are not applicable.

Recently, two general methods for scaling up instance selection algorithms have been published in the literature: stratification and democratization. Both methods are able to successfully deal with large datasets. In this paper we show a comparison of these two methods when applied to very large and huge datasets up to 50,000,000 instances. Additionally, we also test their performance in huge datasets that are also class-imbalanced. The comparison is made using a parallel implementation of both methods to fully exploit their possibilities.

Although both methods show very good behavior in terms of testing error, storage reduction and execution time, democratization proves an overall better performance.

1 Introduction

The overwhelming amount of data that is available nowadays in any field of research poses new problems for data mining and knowledge discovery methods. This huge amount of data makes most of the existing algorithms inapplicable to many real-world problems. Data reduction consists of removing missing, redundant and/or erroneous data to get a tractable amount of information. One of the most common methods for data reduction is instance selection.

Instance selection [8] consists of choosing a subset of the total available data to achieve the original purpose of the data mining application as successfully as the purpose would have been achieved with the whole dataset.

* This work was supported in part by the Project TIN2008-03151 of the Spanish Ministry of Science and Innovation and the project P09-TIC-4623 of the Junta de Andalucía

We can distinguish two main models [2]: instance selection as a method of prototype selection for algorithms based on prototypes (such as k -nearest neighbors) and instance selection for obtaining the training set for a learning algorithm that uses this training set (such as classification trees or neural networks).

The problem of instance selection for instance based learning can be defined as [1] “the isolation of the smallest set of instances that enable us to predict the class of a query instance with the same (or higher) accuracy than the original set”.

García-Osorio et al. [6] proposed an algorithm called *democratic* instance selection that was able to achieve a large reduction in the execution time of the instance selection algorithms while keeping their performance. The underlying idea was to follow the philosophy of classifier ensembles and carry out several rounds of weak instance selection algorithms and combine them using a voting scheme. Therefore, this approach was called *democratic* instance selection.

Democratic instance selection is thus based on repeating several rounds of a fast instance selection process. Each round on its own would not be able to achieve a good performance. However, the combination of several rounds using a voting scheme is able to match the performance of an instance selection algorithm applied to the whole dataset with a large reduction in the time of the algorithm. Thus, in a different setup from the case of ensembles of classifiers, we can consider this method a form of “ensembling” instance selection.

In this paper, we use a parallel implementation of this method that is able to achieve a tremendous reduction in the execution time of any instance selection algorithm while keeping its performance. A further advantage is the reduction in memory storage requirements. The algorithm does not need to have in memory the whole dataset. For huge problems where we have millions of instances it means that we can perform the instance selection when other algorithms would be limited by the amount of available memory.

Another very efficient approach that has been recently proposed is the stratification strategy[3]. In this paper, we compared parallel implementations of these two methods in a set of benchmarks problems that can be considered large or very large datasets. The aim of the comparison is to test three basic hypothesis:

1. As both methods proved their usefulness for scaling up instance selection methods in medium and large datasets, we want to test whether this good performance is kept in very large datasets. As standard instance selection methods cannot be applied due to their computational cost, we will test the performance of the scaling up methods using as base measure the nearest neighbor error using the whole dataset.
2. As democratization is more computationally expensive, we want to test whether this added complexity pays off with an improvement in the performance.
3. Class-imbalanced problems are becoming relevant in many fields of research of machine learning. We want also to test the behavior of both methods when facing this kind of problems.

This paper is organized as follows: Section 2 presents the two methods that are being compared; Section 3 shows their parallel implementation; Section 4 shows the results of the experiments; and Section 5 states the conclusions of our work and future research lines.

2 Democratization and Stratification of Instance Selection Methods

The stratification strategy splits the training data into disjoint strata with equal class distribution¹. The training data, T , is divided into t disjoint datasets, D_j , of approximately equal size:

$$T = \bigcup_{j=1}^t D_j. \quad (1)$$

Then, the data mining algorithm is applied to each subset separately and the results of all of the subsets are combined for the final solution. If we have an algorithm of quadratic complexity, $O(N^2)$, for a number of instances, N , and we employ M strata, we will have a time complexity $O(N/M)^2$. Because we have to apply the method to the M strata, the resulting complexity is $O(N^2/M)$, which is M times faster than the original algorithm. If we are able to run the algorithm in parallel in all the strata, our complexity will be $O(N^2/M^2)$ with a speedup of M^2 . Of course, the drawback of this approach is the likely decrease in the performance of the algorithm. Derrac et al.[4] used stratification to scale up a steady-state memetic algorithm for instance selection. These results support the fact that a sophisticated algorithm applied in a suboptimal way to allow the scaling up of the method can still improve the results of less complex methods applied to the whole dataset.

Democratization shares the data partitioning philosophy of stratification. The democratization process consists of dividing the original dataset into several disjoint subsets of approximately the same size that cover all the dataset. Then, the instance selection algorithm is applied to each subset separately. The instances that are selected to be removed by the algorithm receive a vote. Then, a new partition is performed and another round of votes is carried out. After the predefined number of rounds is made, the instances which have received a number of votes above a certain threshold are removed. Each round can be considered to be similar to a classifier in an ensemble, and the combination process by voting is similar to the combination of base learners in bagging or boosting.

An important step is partitioning the training set into a number of disjoint subsets, t_i , which comprise the whole training set, $\bigcup_i t_i = T$. The size of the subsets is fixed by the user. The actual size has no relevant influence over the results provided is small enough to avoid large execution time. Furthermore,

¹ When dealing with class-imbalance problems the distribution of classes in each strata can be modified with respect to the whole training set to obtain a less skewed distribution.

the time spent by the algorithm depends on the size of the largest subset, so it is important that the partition algorithm produces subsets of approximately equal size.

In the same way, stratification performs a stratified partition of the dataset. To avoid any advantage of any of the two methods we have performed the same kind of stratified partition for both methods. The partition is carried out randomly, keeping approximately, the same distribution of classes as in the whole dataset.

2.1 Determining the Number of Votes

An important issue in democratization is determining the number of votes needed to remove an instance from the training set. It is not possible to set a general preestablished value usable in any dataset. On the contrary, we need a way of selecting this value directly from the dataset in run time. The method to obtain this threshold is based on estimating the best value for the number of votes from the effect on the training set. The election of the number of votes must take into account two different criteria: training error, ϵ_t , and storage, or memory, requirements m . Both values must be minimized as much as possible. The method of choosing the number of votes needed to remove an instance is based on obtaining the threshold number of votes, v , that minimizes a fitness criterion, $f(v)$, which is a combination of these two values:

$$f(v) = \alpha\epsilon_t(v) + (1 - \alpha)m(v), \quad (2)$$

We perform r rounds of the algorithm and store the number of votes received by each instance. Then, we must obtain the threshold number of votes, v , to remove an instance. This value must be $v \in [1, r]$. We calculate the criterion $f(v)$ (eq. 2) for all the possible threshold values from 1 to r , and assign v to the value which minimizes the criterion. After that, we perform the instance selection removing the instances whose number of votes is above or equal to the obtained threshold v . In this way, the evaluation of each threshold of votes is also democratized.

3 Parallel Implementation

The parallel implementation is based on a master/slave architecture. The master performs the partition of the dataset and sends the subsets to each slave. Each slave performs the instance selection algorithm using only the instances of its subset and then returns the selected instances to the master. The master stores the votes for each removed instance and perform a new round. The general architecture of the system is shown in Figure 1. As each round is independent of the previous one all of them are performed in parallel. This method has the advantage that it is still applicable for huge datasets, as only a small part of the dataset must be kept in memory.

The threshold of votes is obtained using the same parallel *democratic* approach. Again, we divide the dataset into disjoint subsets and evaluate the application of each threshold on every subset separately. The value of the goodness of a threshold is the average value of evaluating eq. 2 in each subset.

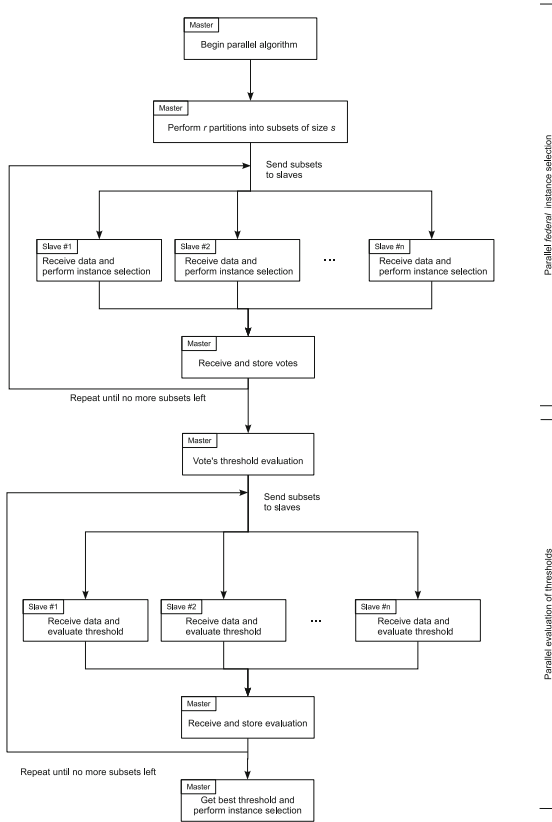


Fig. 1. Parallel implementation of *democratic* instance selection

Stratification is basically the application of democratization just one round. Thus, the same architecture is used for implementing stratification, removing the threshold optimization that is not needed in this method.

4 Experimental Results

We have used a set of eight problems where the number of instances is very large, or even huge. Seven of the datasets are from the UCI Machine Learning Repository, `dna` dataset is from the Pattern Analysis, Statistical Modeling and Computational Learning (PASCAL) challenge². For estimating the storage reduction and generalization error we used 10-fold cross-validation. The source code, in C and licensed under the GNU General Public License, used for all methods as well as the partitions of the datasets are freely available upon request to the authors. Table 1 shows the characteristics of these problems. `census`,

² <http://pascallin2.ecs.soton.ac.uk/Challenges/>

chrom21 and **dna** are class-imbalanced datasets. Their minority/majority class ratio is shown in the table. Testing error, and G -mean value for class-imbalanced datasets are also shown in the table.

We must mention that these datasets are a demanding challenge for any scaling up method. Especially hard are **chrom21** and **dna** problems. **chrom21** has many features, as well as more than one million instances, and it is heavily class-imbalanced. **dna** has 50 millions instances and 800 features. None of these two problems can be addressed without data partitioning as the main memory will be exhausted even by a small percentage of their instances.

Table 1. Summary of datasets. The features of each dataset can be C(continuous), B(binary) or N(nominal). The Inputs column shows the number of input variables after transforming binary and nominal variables to numerical values. For class-imbalanced datasets the minority/majority class ratio is also shown.

Data set	Cases	1-NN error	G -mean	Features			Classes	Inputs	Imbalance ratio
				C	B	N			
census	299,285	—	0.6130	7	—	30	2	409	1/15
chrom21	1,267,701	—	0.0000	—	—	403	2	1612	1/4912
covtype	581,012	0.3024	—	54	—	—	7	54	—
dna	50,000,000	—	0.1580	—	—	200	2	800	1/344
kddcup99	494,021	0.0006	—	33	4	3	23	119	—
kddcup991M	1,000,000	0.0002	—	33	4	3	23	119	—
kddcup99all	4,898,431	0.0002	—	33	4	3	23	119	—
poker	1,025,010	0.4975	—	5	—	5	10	25	—

All the experiments have been carried out in a cluster of 32 blades. Each blade is a bi-processor DELL Power Edge M600 with four cores per processor. Thus, we count with 256 cores. The blades are interconnected with a master node and among them with a 1 Gb network. In all the parallel implementations, we use a master/slave model, where all the information processed by the slaves is sent by the master. The processors run at 2.5 GHz and each blade has 16 Gb of memory.

The execution time shown in the tables is the wall-clock time expended by the algorithms, including every part of them. We measure the time elapsed since the program is started until the program finishes with its final output. That means the time needed to read the data files, perform the partition, send the data to the slaves, receive the results and obtain the best threshold of votes is included in the reported time.

Both methods can be applied using any instance selection algorithm. Cano et al. [2] performed a comprehensive comparison of the performance of different evolutionary algorithms for instance selection and found that evolutionary based methods were able to outperform classical algorithms in both classification accuracy and data reduction. Among the evolutionary algorithms, CHC was able to achieve the best overall performance. Thus, this algorithm has been chosen as our base method. CHC stands for *Cross generational elitist selection, Heterogeneous recombination and Cataclysmic mutation* [5].

Furthermore, the major problem addressed when applying genetic algorithms to instance selection is scaling up the genetic algorithm. As the number of instances grows, the time needed for the genetic algorithm to reach a good solution

increases exponentially, making it totally useless for large problems. In this way, using CHC as instance selection algorithm provides the most extreme scaling up case, a very time consuming algorithm and huge datasets.

We used a standard fitness function based on reduction, r_i , and training error, e_i :

$$f_i = \alpha(1 - e_i) + (1 - \alpha)r_i, \quad (3)$$

where $0 < \alpha < 1$. In the reported experiments $\alpha = 0.5$, which means that reduction and accuracy are considered in equal terms. To evaluate e_i , we use a 1-NN classifier. The training error for instance \mathbf{x} using an individual that selects a subset of instances S is evaluated using as prototype set $S \setminus \{\mathbf{x}\}$.

Accuracy is not a useful measure for class-imbalanced data, especially when the number of instances of the minority class is very small compared to the majority class. If we are concerned about the performance on both negative and positive classes the G – mean measure [7] is a useful value:

$$G - \text{mean} = \sqrt{\text{Specificity} \cdot \text{Sensitivity}}. \quad (4)$$

This measure is a good compromise between specificity and sensitivity. Due to the use of different accuracy measures we shown separately the results for class-imbalanced datasets.

For CHC we used a value of $k = 1$, a population of 100 individuals evolved for 1000 generations with a mutation probability of 10% and a bit mutation probability of 10%. A mutation based on Reduced Nearest Neighbor (RNN) rule was applied with a probability of 5%. These same values are used each time any of these methods is applied, both using stratification and democratization.

For the application of democratization, we used subsets of $s = 1000$ instances, and performed $r = 10$ partitions. The same subset size was used for stratification. Table 2 shows the results which are illustrated in Figure 2. The first remarkable result is the excellent behavior of both approaches in scaling up the evolutionary instance selection algorithm. Even for a dataset with almost 5 million instances and 119 features the execution time is less than an hour for the slowest algorithm. Any estimation of the time needed for a standard CHC algorithm for this dataset would be in the thousands of hours³. Furthermore, the scalability is achieved while keeping the performance of the algorithms. For `covtype`, there is an increment in the testing error, but not very large, and for the remaining datasets the testing error is similar to the error obtained with all the instances, while a very significant storage reduction is achieved.

The results also show that, although stratification is faster, the differences with democratization are not as large as expected. In the worst case, democratization is only 6 times slower than stratification. This is an effect of the ability of democratization of performing the different rounds of the algorithm

³ As an example, standard CHC needed 25 hours for `adult` dataset, which has only 48,842 instances.

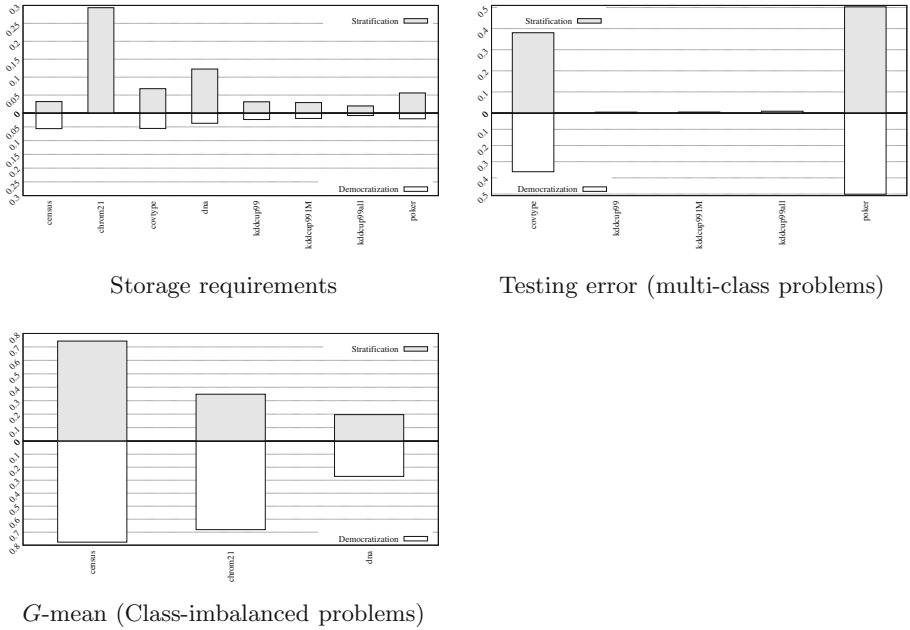


Fig. 2. Storage requirements and error results for stratification and democratization

simultaneously. Storage reductions are very similar with a small advantage of democratization. The same can be said about testing error.

We have performed a statistical test to assure whether the observed differences are statistically significant. As we are comparing results of two methods on a small set of problems using 10-fold cross-validation, we have chosen the corrected resampled t -test [9]. We have compared the testing error and storage requirements of both methods for each problem. Table 2 marks when a method is significantly better at a confidence level of 95% with an *. This test shows that democratization is significantly better for `kddcup99all`, in both storage and error, for `covtype`, in error, and for `poker`, in storage.

Results for class-imbalanced datasets are shown in Table 3. The CHC algorithm applied to class-imbalanced problems was the same that in the previous set of problems with the only modification of substituting the accuracy by the G -mean in the fitness function. In this case, both methods improve the accuracy of 1-NN rule with all the instances. For this kind of problems, the results are somewhat different. In this case, democratization is superior to stratification for all the three datasets and in both testing error and storage reduction. It seems that the combination process added by the democratization approach is able to cope with the increased complexity of the class-imbalanced problems better than stratification.

Table 2. Summary of results for democratization and stratification methods using a CHC algorithm as base method

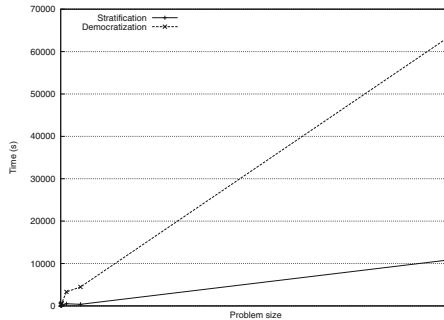
Dataset	Democratization			Stratification		
	Storage	Error	Time (s)	Storage	Error	Time (s)
covtype	0.0556	0.3623*	303.1	0.0678	0.3805	55.9
kddcup99	0.0232	0.0009	335.7	0.0312	0.0040	50.6
kddcup991M	0.0192	0.0007	671.2	0.0290	0.0044	91.7
kddcup99all	0.0086*	0.0003*	3307.0	0.0196	0.0084	520.0
poker	0.0205*	0.5012	497.4	0.0558	0.5031	66.0

Table 3. Summary of results for democratization and stratification methods using a CHC algorithm as base method for class-imbalanced problems

Dataset	Democratization			Stratification		
	Storage	G-mean	Time (s)	Storage	G-mean	Time (s)
census	0.0563*	0.7766*	287.1	0.0318	0.7435	55.9
chrom21	0.0000*	0.6805*	4472.0	0.2937	0.3481	50.6
dna M	0.0368*	0.2707*	63797.0	0.1225	0.1966	91.7

4.1 Time Complexity

The theoretical time complexity of both methods was shown in Section 2. Thus, both methods must show a computational cost that is linear in the number of instances, or even constant if we have enough processors. To illustrate this property, we show the behavior of both methods in terms of execution time in function of the number of instances in Figure 3. We plot the time spent by the algorithms as the complexity of the problem increases. This corroborates the theoretical arguments given above. The figure also shows that the constant term influences the time complexity of democratization with a steeper curve.

**Fig. 3.** Computational cost, in logarithmic scale, of our method and a base instance selection algorithm of $O(n^2)$ and 256 processors

5 Conclusions and Future Work

In this paper we have compared two new methods for scaling up instance selection algorithms that are applicable to any instance selection method without any modification. Both methods have shown their ability to scale up instance selection algorithms to very large datasets up to 50,000,000 instances.

The experiments have shown that in terms of execution time, stratification is faster than democratization, although both methods showed linear time complexity. In terms of reduction of storage requirements and testing error, democratization is better than stratification, but the differences are not very large in multi-class balanced problems.

We have also shown a comparison of both methods in class-imbalanced problems. The tested problems present a high imbalance ratio, with a worst case of 1/4912. For these problems, there is clear advantage of democratization over stratification in terms of both storage reduction and accuracy, measured using G -mean value. This last result opens the possibility of reformulating democratization for the specific case of class-imbalanced problems.

References

1. Brighton, H., Mellish, C.: Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery* 6, 153–172 (2002)
2. Cano, J.R., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. *IEEE Transactions on Evolutionary Computation* 7(6), 561–575 (2003)
3. Cano, J.R., Herrera, F., Lozano, M.: Stratification for scaling up evolutionary prototype selection. *Pattern Recognition Letters* 26(7), 953–963 (2005)
4. Derrac, J., García, S., Herrera, F.: Stratified prototype selection based on a steady-state memetic algorithm: a study of scalability. *Memetic Computing* 2, 183–189 (2010)
5. Eshelman, L.J.: The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. Morgan Kaufman, San Mateo (1990)
6. García-Osorio, C., de Haro-García, A., García-Pedrajas, N.: Democratic instance selection: a linear complexity instance selection algorithm based on classifier ensemble concepts. *Artificial Intelligence* 174, 410–441 (2010)
7. Kubat, M., Holte, R., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. *Machine Learning* 30, 195–215 (1998)
8. Liu, H., Motoda, H.: On issues of instance selection. *Data Mining and Knowledge Discovery* 6, 115–130 (2002)
9. Nadeau, C., Bengio, Y.: Inference for the generalization error. *Machine Learning* 52, 239–281 (2003)

C4.5 Consolidation Process: An Alternative to Intelligent Oversampling Methods in Class Imbalance Problems

Iñaki Albisua, Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, and Jesús M. Pérez

Computer Science Faculty, University of the Basque Country,

Manuel Lardizabal 1, 20018 Donostia, Spain

{inaki.albisua, olatz.arbelaitz, i.gurrutxaga, j.muguerza,
txus.perez}@ehu.es

<http://www.sc.ehu.es/aldapa>

Abstract. In real world problems solved using data mining techniques, it is very usual to find data in which the number of examples of one of the classes is much smaller than the number of examples of the rest of the classes. Many works have been done to deal with these problems known as class imbalance problems. Most of them focus their effort on data resampling techniques so that training data would be improved, usually balancing the classes, before using a classical learning algorithm. Another option is to propose modifications to the learning algorithm. As a mixture of these two options, we proposed the Consolidation process, based on a previous resampling of the training data and a modification of the learning algorithm, in this study the C4.5. In this work, we experimented with 14 databases and compared the effectiveness of each strategy based on the achieved AUC values. Results show that the consolidation obtains the best performance compared to five well-known resampling methods including SMOTE and some of its variants. Thus, the consolidation process combined with subsamples to balance the class distribution is appropriate for class imbalance problems requiring explanation and high discriminating capacity.

Keywords: class imbalance problems, resampling methods, SMOTE, consolidation process, C4.5 decision trees.

1 Introduction

Information technologies allow us to store large amounts of data. This data is combined with data mining techniques in different areas such as medicine, marketing, fraud detection in insurance companies, etc. to extract useful information from it. In some of these problems the number of examples of a class is much smaller than the number of examples of the rest of the classes –highly skewed datasets or class imbalance problems. In these problems the instances are usually grouped into the majority or negative class and the minority class or positive class and it is usual the less representative class to be the most interesting one and the one with bigger cost when making errors from the point of view of learning. This is one of the emergent challenges in Data Mining in the last years [28].

Most machine learning algorithms are designed so that the built classifier reduces the classification error. This means that if we face a problem where the class distribution is 0.1%-99.9% (lets' think for example, in a problem where the aim is to detect a very odd illness), a classifier that classifies every example as healthy patients will be right in 99.9% of the examples. However, this classifier won't have extracted any knowledge about the problem it needed to solve. The same problem can be found in many other application areas such as detection of fraudulent customers in an insurance company, bank credit award, etc.

Many approaches have been proposed to deal with class imbalance problems but they can be divided in two main groups: algorithmic approaches and data approaches. The first ones propose modifications in the algorithms such as the improved boosting proposed by Joshi et al. [17], the modification of SVM proposed by Wu et al [11], an alternative based on one-class learning [19], and some more options as can be seen in [21][9][29]. The other option, data approaches, usually resample (subsample or oversample) the data before using a classical learning algorithm. Most of the research efforts have been done in this direction and they try to balance the class distribution before learning the classifier [6][2][5][16][10]. One of the most popular techniques in this sense is SMOTE [7]: an intelligent oversampling technique to generate more minority class examples. A wide analysis and comparison of some variants can be found in [4][13].

A third option to solve class imbalance problems would be a combination of the previous ones: data and algorithmic approaches. We can find an example of this approach in [10].

In this context, in order to obtain accurate classifiers able to solve class imbalance problems, in [22] the authors proposed to use several samples to build a single decision tree. The method, called CTC –Consolidated Tree Construction Algorithm– is based on a decision tree construction algorithm, C4.5, but it extracts knowledge from data using a set of samples instead of a single one as C4.5 does. In contrast to other methodologies based on several samples to build a classifier, such as bagging, the CTC builds a single tree and as a consequence, the CTC algorithm obtains comprehensible classifiers called Consolidated Trees or CT trees.

Taking into account the context presented in the introduction, in this work we will try to answer the following research question: Is the consolidation a good way to solve class imbalance problems? Does it improve the results achieved with the original sample? And, does this option work better than intelligent oversampling techniques such as SMOTE and its variants?

To answer these questions we performed experiments with 14 two-class real problems and different strategies to solve the class imbalance problem: the CTC (or consolidated C4.5), random subsampling, random oversampling, and 3 options of intelligent oversampling (SMOTE [7], borderline-SMOTE1 [15], borderline-SMOTE2[15]). Furthermore, we run a 10 fold cross-validation methodology five times to estimate the generalization capacity of the built classifiers. We used AUC (Area Under the ROC curve) as a performance estimator. Finally we applied the statistical tests proposed by Demšar [8] and Garcia et al in [14] to evaluate statistical significance of the results.

The paper proceeds in Section 2 by briefly introducing the C4.5 and the consolidation process: the CTC. Section 3 describes the used intelligent oversampling methods used in this work. Section 4 is devoted to describe the experimental methodology. We present a summary of the experimental results and the major outcomes in Section 5. Finally, Section 6 is devoted to conclusions and further work.

2 C4.5 and the Consolidation Process

The C4.5 [24] achieves high quality results as single classifier with an added value: it provides with explanation the proposed classification. It was identified as one of the top 10 algorithms in data mining in the IEEE International Conference in Data Mining held in 2006 [27]. And furthermore, in a recent work Garcia and Herrera [12] proposed very powerful statistical tests for significance analysis and compared the results of five widely used classification algorithms –C4.5, NaiveBayes, CN2, 1-NN and a Kernel classifier– in 30 databases. The use of the tests showed that C4.5 was the algorithm with highest effectiveness followed by NaiveBayes, without significant differences, and, 1-NN and the Kernel classifier with significant differences. Moreover, the C4.5 is widely used after applying oversampling techniques in class imbalance problems [7][4][13].

The consolidation process of the C4.5, was designed to solve a class imbalance problem, a fraud detection problem. The aim was to build a single tree (a classifier with explaining capacity) but using a set of samples with modified class distribution. This process was called CTC (Consolidated Tree Construction) algorithm [22][23].

The CTC algorithm uses the main idea of bagging: voting. In bagging a set of classifiers is built and, then, voting is used to classify new examples. Nevertheless, the CTC algorithm uses a voting process to decide the variable that will be used to split the node at each step of the tree's building process. The decision is made based on different samples. The repetition of this process in every node leads to the construction of a single tree. Although the idea of CTC is wider, CT trees were built based on the standard C4.5 classification algorithm. In order to make the CTC comparable to the C4.5, the split function used is the gain ratio criterion; used by Quinlan in the C4.5 [24]. This idea was kept in the different steps of the algorithm; using the default parameters of the C4.5.

The algorithm starts with the subsampling phase: extracting a set of *Number_Sample* (N_S) samples from the original training set based on the desired resampling technique (R_M). Each node of the CT tree is consolidated based on the information, or split proposals (feature + branches), generated from N_S samples. In the consolidation phase, the feature and branches are selected after a voting process based on all the proposals. And then, all the nodes are forced to split based on this decision. In the Consolidated Tree's generation process nodes stop splitting or do not generate branches when most of the proposals are not to split it, so, to become a leaf node (stopping criteria). The a posteriori probabilities of the leaves are calculated by averaging the a posteriori obtained from the data partitions related to that leaf node in all the samples.

3 Used Intelligent Oversampling Methods

3.1 SMOTE

SMOTE (Synthetic Minority Oversampling TEchnique) [7] is an oversampling algorithm where the minority class is oversampled to generate new synthetic examples. The basic idea is to generate new examples that are located somewhere in the line that sticks together each of the minority class examples and some (or all) of its k nearest neighbours. The distance between examples is usually calculated based on Euclidean distance, however Euclidean distance is not adequate for qualitative (not quantitative) features. The calculation for those cases is usually done using a replacement or overlap function that assigns a value equal to 0 (when both values are the same) or equal to 1 (if they are different). We have implemented SMOTE using HVDM (Heterogeneous Value Difference Metric) distance [26] which uses Euclidean distance for quantitative attributes and VDM distance for the qualitative ones. The VDM metric is more adequate than the overlap metric since it takes into account the similarities among the possible values of each qualitative attribute to calculate the distances between them. Finally, missing values are usually replaced by the average of that attribute for the rest of examples of its class in the case of quantitative attributes. For qualitative attributes the mode is used.

The synthetic examples are generated with the following procedure: calculate the difference between the feature vector of the current example (a minority class example) and the feature vector of one of its nearest neighbours, randomly selected. Then, multiply the difference vector with a random value between 0 and 1 and finally, add this vector to the feature vector of the current example. The new vector will be the synthetic example.

The number of times that a neighbour has to be selected to be used to generate a new example depends on the number of new examples we must generate. For example, if we need to duplicate the number of examples in the minority class it will be enough to use a neighbour for each of the minority class examples.

3.2 Borderline-SMOTE

The main difference of these methods with SMOTE is that only the minority examples in the borderline are oversampled. The minority class examples will be considered to be in the borderline if more than half of their k nearest neighbours belong to the majority class. That is, those examples located in the frontier of the majority and minority class.

The authors propose two different approaches [15]: Borderline-SMOTE1 and Borderline-SMOTE2. The Borderline-SMOTE1 option uses just the minority class neighbours of the examples in the borderline to generate the synthetic examples. However the Borderline-SMOTE2 option uses all the neighbours (minority and majority class) of the examples in the borderline to generate the synthetic examples. If the selected neighbour belongs to the majority class, the random value generated to multiply the difference vector will be in the range 0 - 0.5.

4 Methodology for Experiments

We performed experiments with 14 real problems from the UCI Repository benchmark [3]. These 14 databases are part of the 30 datasets Albisua et al used in a previous work [1], just those with an original minority class distribution below 30%. The characteristics of the databases (see Table 1) vary from 155 examples to 5,620 examples, from 6 to 64 features and from 2 to 15 classes, although for the experiments, we converted all the databases in two-class databases designating the least frequently occurring class as the minority class, and then, mapping the remaining classes into the majority class. And finally, the minority class distribution goes from 3.45 to 30, being the average 18.33%.

Table 1. Domains used in the experimentation and their characteristics

<i>Domain</i>	<i>N. of patterns</i>	<i>N. of features</i>	<i>N. of classes</i>	<i>Minority class</i>	<i>missing values</i>
<i>Soybean-Large</i>	290	35	<u>15</u>	<u>3.45</u>	y
<i>Hypo</i>	3 163	25	2	4.77	y
<i>Sick_euthyroid</i>	3 120	25	2	9.26	y
<i>Optdigits</i>	<u>5 620</u>	<u>64</u>	10	9.9	n
<i>Segment210</i>	210	19	7	14.29	n
<i>Segment2310</i>	2 310	19	7	14.29	n
<i>Kddcup</i>	4 941	41	2	19.69	n
<i>Hepatitis</i>	<u>155</u>	19	2	20.65	y
<i>Vehicle</i>	846	18	4	23.52	n
<i>Glass</i>	214	9	7	23.83	n
<i>Splice_junction</i>	3 190	60	3	24.1	n
<i>Yeast</i>	1 484	8	10	28.9	n
<i>Credit-g</i>	1 000	20	2	30	n
<i>Car</i>	1 728	<u>6</u>	4	<u>30</u>	n

We used a 10-fold cross-validation methodology five times (5x10CV) to estimate the generalization capacity of the built classifiers. As a consequence, we obtained 50 pairs of training and test samples for each database. From each training sample, we further generated 100 balanced (50%) samples using random subsampling and following Weiss and Provosts' proposal [25], 50 samples using SMOTE, 50 samples using Borderline-SMOTE1, 50 samples using Borderline-SMOTE2, and finally, 50 samples using random oversampling to be used as baseline for the intelligent oversampling methods. Thus, in this experiment, we used 5 resampling methods and generated 5x10x300 samples per database.

In order to generate the 100 balanced subsamples we repeated the subsampling process Weiss and Provost did in their work and fixed the size of the subsamples to the number of examples of the minority class: that is we undersampled both the majority and the minority class. If we take into account the 14 databases, the average size of the generated samples is 18.33% of the training samples (the average percentage of the minority classes). To generate samples generated using the different options for SMOTE or random oversampling were also balanced samples but they used all the majority class examples and oversampled the minority class to achieve samples with 50% class distribution. As a consequence the size of the used samples was 145% of the training sample.

We built a C4.5 tree with each one of the samples obtained oversampling the training sample with the different options and the subsamples obtained undersampling it. We also built another C4.5 tree with the whole training sample of the fold without using any sampling method. On the other hand, we used the 100 samples generated under-sampling the training sample, to build all the possible CT trees with disjoint sets of 30 samples. That is, 3 CT trees with $N_S=30$. Although the optimal value for N_S depends on the concrete data base, previous works have proven that 30 is in general a good value [22][23].

On the other hand, as Weiss and Provost [25] proposed in their work, in the evaluation process we corrected the a posteriori probabilities of the leaf nodes to adequate them to the class distribution found in the original training set. They named this correction factor oversampling ratio.

We pruned all the trees, C4.5 and CT trees, using the default pruning proposed by Quinlan [24]. This was the option where C4.5 had the best performance.

We tested the built trees with the test sample in each fold and we measured the obtained AUC (Area Under the ROC Curve) because we agree with other authors such as Ling et al. [18] and Marrocco et al. [20] that AUC is a better measurement than accuracy in comparing learning algorithms when the class distribution changes.

In addition, we used the non-parametric statistical tests proposed by Demšar in [8] and García et al in [14] to evaluate the statistical significance of the results.

5 Experimental Results

As described in the experimental setup section, we performed extensive experiments to compare the results obtained with the mentioned resampling methods and the consolidation process. In the next table, Table 2, we summarize the results for each of the seven strategies used for the 14 databases.

The column ORIG shows the AUC values related to the non-resampling option (the original class distribution). The columns SUB and OVER show the results for the random subsampling and oversampling methods. SMT, B_SMT1 and B_SMT2 refer to SMOTE, Borderline-SMOTE1 and Borderline-SMOTE2, and finally, CTC shows the consolidation process' performance.

The last two rows in the table show the average AUC obtained (in bold) by each strategy and its average rank, which will be used to make the evaluation of statistically significant differences. As it can be observed, the consolidation process achieved the best rank and average AUC, followed by all of the oversampling methods which outperformed the non-resampling option (ORIG). The worst option was the random subsampling method, being the only option that didn't outperform ORIG, though it could be due to the smaller size of the samples which was in average 18.33% of the training samples (the average percentage of the minority classes).

In order to analyze the statistically significant differences between the obtained results, as explained before, we applied the tests proposed by Demšar in [8] and García et al in [14] for multiple classifier comparisons. Iman and Davenport's test showed that there were significant differences between the seven strategies compared. The FF statistic obtained was in this case 4.3165, which was greater than the required value with a 95% significance level (2.2172).

Table 2. AUC values for 7 strategies and 14 databases

Domain	ORIG	SUB	OVER	SMT	B_SMT1	B_SMT2	CTC
<i>Soybean-Large</i>	82.97	82.19	90.20	89.82	89.76	91.63	89.36
<i>Hypo</i>	96.05	95.96	96.31	96.01	96.22	95.52	96.81
<i>Sick_euthyroid</i>	93.76	94.32	94.65	93.98	94.70	94.00	95.01
<i>Optdigits</i>	84.47	88.01	92.14	90.29	89.28	89.53	94.28
<i>Segment210</i>	98.65	97.91	98.96	99.20	99.18	99.21	98.83
<i>Segment2310</i>	94.83	88.68	91.07	92.00	91.39	95.16	93.83
<i>Kddcup</i>	99.39	98.99	99.40	99.35	99.43	99.53	99.54
<i>Hepatitis</i>	66.58	71.53	72.44	66.71	66.87	77.20	68.43
<i>Vehicle</i>	92.88	92.78	92.47	93.86	93.86	94.06	94.91
<i>Glass</i>	88.98	90.06	90.98	91.56	89.93	92.48	92.44
<i>Splice_junction</i>	96.60	94.63	94.64	95.24	95.51	94.33	96.07
<i>Yeast</i>	72.36	72.18	68.09	72.86	72.01	71.92	72.25
<i>Credit-g</i>	65.65	65.36	62.92	67.08	66.18	63.01	68.28
<i>Car</i>	97.85	93.71	97.83	97.70	98.14	98.03	97.99
Average	87.93	87.59	88.72	88.98	88.75	89.69	89.86
Average ranks	4.79 (6)	5.71 (7)	4.29 (5)	3.93 (4)	3.64 (3)	3.29 (2)	2.36 (1)

Once we knew there were significant differences over the whole multiple comparison, we applied several powerful post-hoc procedures proposed by García et al in [14] for 1 x N comparisons, using CTC algorithm as the control method. All of them gave the same qualitative results, as can be seen in Table 3 where results for various tests are shown.

As it can be observed, for a 95% significance level, the tests show that there are significant differences with ORIG and SUB (values marked in bold). To find differences with random oversampling it would be necessary to consider a 90% significance level. Despite achieving better average rank and AUC value than the intelligent oversampling techniques, no significant difference was found by the tests.

Wilcoxon test for comparing two algorithms was also made and significant differences were found between CTC and each of the other options, except Borderline-SMOTE2.

Table 3. Tests' results for *Bonferroni*, *Holm*, *Hochberg* and *Hommel* (1 x N comparisons) and *Wilcoxon* (pairwise comparisons) using CTC as control method

Type of comparison	CTC vs	ORIG	SUB	OVER	SMT	B_SMT1	B_SMT2
Multiple (1 x N)	<i>Bonferroni</i>	0.0176	0.0002	0.1091	0.3257	0.6920	1.5326
	<i>Holm</i>	0.0147	0.0002	0.0727	0.1628	0.2307	0.2554
	<i>Hochberg</i>	0.0147	0.0002	0.0727	0.1628	0.2307	0.2554
	<i>Hommel</i>	0.0147	0.0002	0.0727	0.1628	0.2307	0.2554
Pairwise	<i>Wilcoxon</i>	0.0132	0.0076	0.0355	0.0110	0.0132	0.5936

At last, as Demšar proposed in [8], we made Nemenyi and Bonferroni-Dunn tests, which are less powerful but offer a very clear graphic explanation. The results confirm everything that has been said. Figure 1 shows graphically results for the Nemenyi's and Bonferroni-Dunn's tests based on CD (Critical Difference) diagrams.

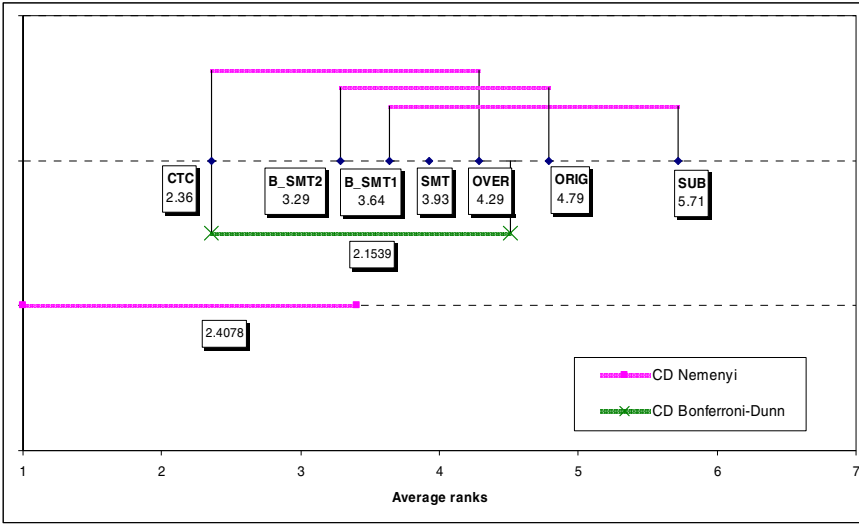


Fig. 1. Results for Nemenyi and Bonferroni-Dunn tests for 7 strategies

In Nemenyi’s test (upper line in the figure) two algorithms are connected by a line if no significant differences exist whereas for Bonferroni-Dunn test (lower line in the figure) there are significant differences with the control classifier (CTC) only if the corresponding point is outside the line. Graphs show that based on both kind of tests, we can come to the same conclusions related with CTC. In terms of statistically significant differences, CTC is the only strategy that outperformed the original. Although having achieved the best ranking value, the differences between consolidation and the rest of the oversampling methods are not significant.

Finally, we would like to emphasize the gain of the consolidation process, especially when this approach has been made based on the worst of the resampling strategies.

6 Conclusions and Further Work

Many works have been done in the context of machine learning when working with imbalanced data sets, most of whom deal with the problem by using data resampling strategies, where SMOTE and some of its variants have become reference methods.

In this work, we have compared some well-known resampling methods (random subsampling, random oversampling, SMOTE, Borderline-SMOTE1 and Borderline-SMOTE2) used to balance the class distribution in the training data, with a strategy that mixes data and algorithmic approach, which we call the consolidation process. All the strategies were applied to the C4.5 algorithm.

We performed an extensive experimentation and applied statistical tests to determine whether there were statistically significant differences between the proposed strategies or not. Results confirmed that not only consolidation is a good way to solve class imbalance problems, but it also works better than the mentioned

intelligent oversampling techniques, in addition to improving the results achieved with the original sample.

After seeing these satisfactory results, there are many things that can be done in the future. Firstly, the experimentation could be extended to more databases and resampling methods combined with cleaning techniques (e.g. SMOTE-ENN). Furthermore, we would like to analyze class distributions different to 50% in order to find the best one for the different resampling techniques. Finally, we hope that the combination of the best samples obtained with intelligent resampling techniques and the consolidation process will achieve better results.

Acknowledgments. This work was funded by the University of the Basque Country, general funding for research groups (GIU10/02), by the Science and Education Department of the Spanish Government (TIN2010-15549 project), by the Diputación Foral de Gipuzkoa.

References

1. Albisua, I., Arbelaitz, O., Gurrutxaga, I., Martín, J.I., Muguerza, J., Pérez, J.M., Perona, I.: Obtaining optimal class distribution for decision trees: Comparative analysis of CTC and C4.5. In: Meseguer, P., Mandow, L., Gasca, R.M. (eds.) CAEPIA 2009. LNCS, vol. 5988, pp. 101–110. Springer, Heidelberg (2010)
2. Artís, M., Ayuso, M., Guillén, M.: Modelling different types of automobile insurance fraud behaviour in the Spanish market. *Insurance: Mathematics and Economics* 24, 67–81 (1999)
3. Asuncion, A., Newman, D.J.: UCI ML Repository. University of California, School of Information and Computer Science, Irvine (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
4. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data SIGKDD Explorations Newsletter. *ACM* 6, 20–29 (2004)
5. Berry, M.J.A., Linoff, G.: *Astering Data Mining. The Art and Science of Customer Relationship Management*. Willey (2000)
6. Chan, P.K., Stolfo, S.J.: Toward Scalable Learning with Non-uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. In: Proc. of the 4th. Int. Conf. on Knowledge Discovery and Data Mining, pp. 164–168. AAAI Press, Menlo Park (1998)
7. 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)
8. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
9. Elkan, C.: The Foundations of Cost-Sensitive Learning. In: Proceedings of the 17th. Int. Joint Conf. on Artificial Intelligence, pp. 973–978 (2001)
10. Estabrooks, A., Jo, T.J., Japkowicz, N.: A Multiple Resampling Method for Learning from Imbalanced Data Sets. *Computational Intelligence* 20(1), 18–36 (2004)
11. Wu, G., Chang, E.Y.: Class-Boundary Alignment for Imbalanced Dataset Learning. In: Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC (2003)
12. García, S., Herrera, F.: An Extension on “Statistical Comparisons of Classifiers over Multiple Data Sets” for all Pairwise Comparisons. *Journal of Machine Learning Research* 9, 2677–2694 (2008)

13. García, S., Fernández, A., Herrera, F.: Enhancing the Effectiveness and Interpretability of Decision Tree and Rule Induction Classifiers with Evolutionary Training Set Selection over Imbalanced Problems. *Applied Soft Computing* 9, 1304–1314 (2009)
14. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental Analysis of Power. *Information Sciences* 180, 2044–2064 (2010)
15. Han, H., Wang, W., Mao, B.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning *Advances in Intelligent Computing*, pp. 878–887 (2005)
16. Japkowicz, N., Stephen, S.: The Class Imbalance Problem: A Systematic Study. *Intelligent Data Analysis Journal* 6(5), 429–449 (2002)
17. Joshi, M., Kumar, V., Agarwal, R.: Evaluating Boosting Algorithms to Classify Rare Classes: Comparison and Improvements. In: *First IEEE International Conference on Data Mining*, San Jose, CA (2001)
18. Ling, C.X., Huang, J., Zhang, H.: AUC: A better measure than accuracy in comparing learning algorithms. In: Xiang, Y., Chaib-draa, B. (eds.) *Canadian AI 2003. LNCS (LNAI)*, vol. 2671, pp. 329–341. Springer, Heidelberg (2003)
19. Manevitz, L.M., Yousef, M.: One-class SVMs for document classification. *Journal of Machine Learning Research* 2, 139–154 (2001)
20. Marrocco, C., Duin, R.P.W., Tortorella, F.: Maximizing the area under the ROC curve by pairwise feature combination. *Pattern Recognition* 41(6), 1961–1974 (2008)
21. Orriols-Puig, A., Bernadó-Mansilla, E.: Evolutionary rule-based systems for imbalanced data sets. *Soft Computing* 13, 213–225 (2009)
22. Pérez, J.M., Muguerza, J., Arbelaitz, O., Gurrutxaga, I., Martín, J.I.: Consolidated Tree Classifier Learning in a Car Insurance Fraud Detection Domain with Class Imbalance. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) *ICAPR 2005. LNCS*, vol. 3686, pp. 381–389. Springer, Heidelberg (2005)
23. Pérez, J.M., Muguerza, J., Arbelaitz, O., Gurrutxaga, I., Martín, J.I.: Combining multiple class distribution modified subsamples in a single tree. *Pattern Recognition Letters* 28(4), 414–422 (2007)
24. Quinlan, J.R.: *C4.5 Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Mateo (1993)
25. Weiss, G.M., Provost, F.: Learning when Training Data are Costly: The Effect of Class Distribution on Tree Induction. *Journal of Artificial Intelligence Research* 19, 315–354 (2003)
26. Wilson, D.R., Martínez, T.R.: Reduction Techniques for Exemplar-Based Learning Algorithms. *Machine Learning* 38(3), 257–286 (2000)
27. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, M., Hand, D.J., Steinberg, D.: Top 10 algorithms in data mining. *Knowledge and Information Systems* 14, 1–37 (2008)
28. Yang, Q., Wu, X.: 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making* 5(4), 597–604 (2006)
29. Zadrozny, B., Elkan, C.: Learning and Making Decisions When Costs and Probabilities are Both Unknown. In: *Proceedings of the 7th. Int. Conf. on Knowledge Discovery and Data Mining*, pp. 204–213 (2001)

Scalability Analysis of ANN Training Algorithms with Feature Selection*

Verónica Bolón-Canedo, Diego Peteiro-Barral, Amparo Alonso-Betanzos, Bertha Guijarro-Berdiñas, and Noelia Sánchez-Marcoño

Laboratory for Research and Development in Artificial Intelligence (LIDIA), Computer Science Dept., University of A Coruña, 15071 A Coruña, Spain
{vbolon,dpeteiro,ciamparo,cibertha,nsanchez}@udc.es

Abstract. The advent of high dimensionality problems has brought new challenges for machine learning researchers, who are now interested not only in the accuracy but also in the scalability of algorithms. In this context, machine learning can take advantage of feature selection methods to deal with large-scale databases. Feature selection is able to reduce the temporal and spatial complexity of learning, turning an impracticable algorithm into a practical one. In this work, the influence of feature selection on the scalability of four of the most well-known training algorithms for feedforward artificial neural networks (ANNs) is studied. Six different measures are considered to evaluate scalability, allowing to establish a final score to compare the algorithms. Results show that including a feature selection step, ANNs algorithms perform much better in terms of scalability.

1 Introduction

In the past few years, the increase of computational power, bandwidth and storage capacity has brought an interesting challenge for machine learning researchers, since the huge amount of data now available has led to datasets of high dimensionality which makes the machine learning task more complex and computationally demanding. The term *high dimensionality* means that a database presents several of the following characteristics: (a) the number of *samples* is very high; (b) the number of *features* is very high. Machine learning gets particularly difficult when dealing with datasets with more than 1 000 000 data (where data means *samples* \times *features*), or even much less. The problem here is that practically all machine learning algorithms operate with the training set entirely in main memory and, consequently, their spatial and/or temporal complexity grows as the number of samples increases. In fact, in many cases, learning algorithms are not able to process the whole training set due to time or memory restrictions and, in practice, preprocessing techniques are required.

Within preprocessing techniques, machine learning can take advantage of feature selection methods to be able to confront these problems. Theoretically,

* This work was supported by Spanish Ministerio de Ciencia e Innovación under project TIN 2009-02402, partially supported by the European Union ERDF.

having more data should give more discriminating power. However, the nature of high dimensionality of data can cause the so-called problem of “curse of dimensionality”. To avoid this problem, feature selection plays a crucial role. *Feature selection* (FS) consists of detecting the relevant features and discarding the irrelevant ones in order to reduce the input dimensionality and, most of the time, to achieve an improvement in performance [1]. The benefits of FS have been proven by the authors in diverse fields such as bioinformatics [2] or intrusion detection [3]. In general, there are three different models for feature selection: filter, wrapper and embedded methods. While wrapper models use a specific prediction method as a black box to score subsets of features as part of the selection process, filter models rely on the general characteristics of the training data to select features with independence of any predictor. Halfway these two models one can find the embedded methods, which perform FS as part of the training process of the prediction model. By having some interaction with the classifier, wrapper and embedded methods tend to give better performance results than filters. However, in this work, the high dimensionality of the datasets prevent the application of these methods and only the filter model will be considered.

Large-scale learning [4,5,6], which is located within machine learning, intends to develop efficient and scalable algorithms with regard to requirements of computation, memory, time and communications. Increasing the size of the training set often increases accuracy [7] but, if the computational complexity of the algorithm exceeded the main memory then the algorithm will not scale well or will be unfeasible to run. Thus, for scaling up learning algorithms, the issue is not so much as one of speeding up a slow algorithm as one of tuning an impracticable algorithm into a practical one, i.e. the crucial issue is seldom *how fast* you can run on a particular problem, but rather *how large* a problem can you deal with [8]. Eventually, even if the scalability of learning algorithms is improved, there is still the question of its impact on the own goal of learning. Assessing performances become complicated if a degradation in the quality of learning is permitted, since it is essential to minimize training time and allocated memory while maintaining accuracy. However, up to now, most machine learning algorithms do not provide an appropriate balance among them and tend to look with favor on one of these variables against another. In this work, the authors are most interested in methods that scale up algorithms without a substantial decrease in accuracy.

In a previous work [9], the authors have studied the scalability of several training algorithms for ANNs on large-scale databases, using the measures defined in the workshop *PASCAL Large Scale Learning Challenge* [10] at the 25th International Conference on Machine Learning (ICML’08). These measures assess the scalability of algorithms in terms of error, computational effort, allocated memory and training time. In this work, the influence of feature selection on the scalability of four of the most well-known training algorithms for feedforward ANNs will be studied. By reducing the number of input features and, consequently, the dimensionality of the dataset, we expect to reduce the

computational time while maintaining the performance on the other measures, as well as being able to apply certain algorithms which could not deal with so large databases.

The remainder of this paper is structured as follows: section 2 describes the feature selection process, section 3 presents the experimental study, section 4 shows the experimental results and sections 5 and 6 include the discussion and conclusions, respectively.

2 Feature Selection

Feature selection is the process of detecting the relevant features and discarding the irrelevant ones, with the goal of obtaining a subset of features that describes properly the given problem with a minimum degradation of performance [1]. It has several advantages, such as:

- Improving the performance of the machine learning algorithms.
- Data understanding, gaining knowledge about the process and perhaps helping to visualize it.
- Data reduction, limiting storage requirements and perhaps helping in reducing costs.
- Simplicity, possibility of using simpler models and gaining speed.

In light of the above, FS seems to be helpful in reducing the computational effort, allocated memory and training time, measures that will be considered to study the scalability of the machine learning algorithms.

As was stated in the Introduction, when managing high dimensional datasets, the application of wrappers and embedded methods is not practicable, and therefore our study will be focused in the use of filters. Filters rely on the general characteristics of the training data to select features with independence of any predictor and are usually computationally less expensive than wrappers and embedded methods. Moreover, filters have the ability to scale to large datasets and result in a better generalization because they act independently of the induction algorithm. So, in those cases in which the dimensionality is very high, filter methods are a good choice to obtain a reduced set of features that can be treated by the machine learning algorithms.

With regard to the filter model, there exist two major approaches: *individual evaluation* and *subset evaluation* [1]. Individual evaluation is also known as feature ranking and assesses individual features by assigning them weights according to their degrees of relevance. On the other hand, subset evaluation produces candidate feature subsets based on a certain search strategy. Each candidate subset is evaluated by a certain evaluation measure and compared with the previous best one with respect to this measure. While the individual evaluation is incapable of removing redundant features because redundant

features likely have similar rankings, the subset evaluation approach can handle feature redundancy with feature relevance, so this will be the approach chosen for this study.

Correlation-based Feature Selection (CFS) [12] is one of the most well-known and widely-used filters, following the subset evaluation approach, which has proven to obtain good results on previous works, so it will be chosen for this study. CFS is a simple filter algorithm that ranks feature subsets according to a correlation based heuristic evaluation function [12]. The bias of the evaluation function is toward subsets that contain features that are highly correlated with the class and uncorrelated with each other. Irrelevant features should be ignored because they will have low correlation with the class. Redundant features should be screened out as they will be highly correlated with one or more of the remaining features. The acceptance of a feature will depend on the extent to which it predicts classes in areas of the instance space not already predicted by other features. CFS’s feature subset evaluation function is:

$$M_S = k\overline{r_{cf}} / \sqrt{k + k(k-1)\overline{r_{ff}}},$$

where M_S is the heuristic ‘merit’ of a feature subset S containing k features, $\overline{r_{cf}}$ is the mean feature-class correlation ($f \in S$) and $\overline{r_{ff}}$ is the mean feature-feature intercorrelation. The numerator of this equation can be thought of as providing an indication of how predictive of the class a set of features is; and the denominator of how much redundancy there is among the features.

3 Experimental Study

In order to check the influence of FS on the scalability of machine learning algorithms that can deal with large datasets, four of the most popular training algorithms for ANNs were selected. Two of these algorithms are gradient descent (GD) [15] and gradient descent with momentum and adaptive learning rate (GDx) [15], whose complexity is $O(n)$. The other algorithms are scaled conjugated gradient (SCG) [16] and Levenberg-Marquardt (LM) [17], whose complexities are $O(n^2)$ and $O(n^3)$, respectively.

3.1 High Dimensional Datasets

One of the most common tasks in machine learning is classification, and in this work four large datasets were selected to be classified by the algorithms mentioned above. Table 1 shows the datasets used in this paper along with a brief description of them (number of features, classes, training samples and test samples).

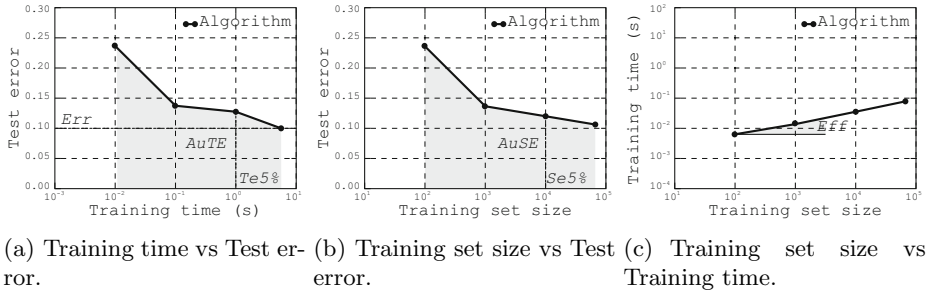
¹ Connect-4 and Covertype datasets are available on <http://archive.ics.uci.edu/ml/datasets.html>; KDD Cup 99 on <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>; and MNIST on <http://yann.lecun.com/exdb/mnist/>

Table 1. Datasets description

Dataset	Features	Classes	Training	Test
Connect-4	42	3	60 000	7 557
KDD Cup 99	42	2	494 021	311 029
Coverttype	54	2	100 000	50 620
MNIST	748	2	60 000	10 000

3.2 Performance Measures

In order to assess the performance of learning algorithms, common measures as accuracy are insufficient since they do not take into account all aspects involved when dealing with large datasets. Accordingly, the goal for machine learning developers is to find a learning algorithm such that it achieves a low error in the shortest possible time using as few samples as possible. Since there are no standard measures of scalability, those defined in the *PASCAL Large Scale Learning Challenge* [10] are used:

**Fig. 1.** Performance measures

- Figure 1(a) shows the relationship between *training time* and *test error*, computed on the largest dataset size the algorithm is able to deal with, aimed at answering the question “Which test error can we expect given limited training time resources?”. Following the PASCAL Challenge, the different training time budgets are set to $10^{[-1,0,1,2,\dots]}$ seconds. We compute the following scalar measures based on this figure:

- *Err*: minimum test error (standard class error [18] for classification).
- *AuTE*: area under Training time vs Test error curve (gray area).
- *Te5%*: the time t for which the test error e falls below a threshold $\frac{e-Err}{e} < 0.05$.

- Figure 1(b) shows the relationship between different *training set sizes* and the *test error* of each one aimed at answering the question “Which test error can be expected given limited training data resources?”. Following the PASCAL Challenge, the different training set sizes (training samples) are set to $10^{[2,3,4,\dots]}$ and the maximum size of the dataset. We compute the following scalar measures based on this figure:
 - *AuSE*: area under Training set size vs Test error curve (gray area).
 - *Se5%*: the size s for which the test error e falls below a threshold $\frac{e-Err}{e} < 0.05$
- Figure 1(c) shows the relationship between different *training set sizes* and the *training time* for each one aimed at answering the question “Which training time can be expected given limited training data resources?”. Again, the different training set sizes are set to $10^{[2,3,4,\dots]}$ and the maximum size of the dataset. We compute the following scalar measure based on this figure:
 - *Eff*: slope b of the curve using a least squares fit to ax^b .

In order to establish a general measure of scalability, the final *Score* of the algorithms is calculated as the average rank of its contribution with regard to the six scalar measures defined above.

3.3 Experimental Settings

During the preprocessing step, the CFS filter was applied over the training set to obtain a subset of features which will be employed in the classification stage, whose result can be seen in Table 2. A significant reduction in the number of necessary inputs can be observed, since after the feature selection process, we will use between a 7% and a 24% of the original features, leading to a reduction in storage requirements and computational cost.

Finally, in order to choose the best training algorithm, different simulations ($N = 10$) were carried out for accurately estimating the scalability of algorithms by applying the following procedure on each dataset after the preprocessing.

Table 2. Reduction after applying feature selection with the CFS filter

Dataset	Original Features	Selected Features	Percentage
Connect-4	42	6	14%
KDD Cup 99	42	5	12%
Coverttype	54	13	24%
MNIST	748	55	7%

Algorithm 1. Experimental procedure

- for $n=1$ to N
 - Divide the original training dataset using *holdout validation*, i.e. a subset of samples is chosen at random to form the validation set and the remaining observations are retained as the training data. The 10% of data is used for testing while the 90% are for training. This kind of validation is suitable because the size of the datasets is large.
 - Set the number of hidden units of the ANN to $2 \times \text{number_of_inputs} + 1$ [19] and train the network. It is important to remark that the aim here is not to investigate the optimal topology of an ANN for a given dataset, but to check the scalability of learning algorithms on large networks.
 - Compute the score of algorithms as the average rank of its contribution with regard to the six scalar measures defined in Section 3.2.
 - Apply a Kruskal-Wallis test to check if there are significant differences among the medians for each algorithm with and without FS for a level of significance $\alpha = 0.05$.
 - If there are differences among the medians, then apply a multiple comparison procedure (Tukey's) to find the simplest approach whose score is not significantly different from the approach with the best score.
-

Notice that the minimum test error (Err) is computed on the test set, whilst the remaining measures on the training set (see Table II).

4 Experimental Results

In this section, the results obtained after assessing the influence of feature selection on the scalability of several learning algorithms over large datasets will be presented. Regarding the performance measures defined in Section 3.2, it has to be noted that *the lower the result, the higher the scalability*.

Table 3 shows the results achieved by applying the CFS filter compared with the results where no feature selection was executed (obtained from a previous work [9]); both studies followed the same methodology and were executed in the same machine for the sake of fairness. Notice that not all the learning algorithms were able to deal with all available samples for every dataset mostly due to the spatial complexity of the algorithms. In particular on the MNIST dataset for which the LM algorithm is not able to train even in the smallest subset. If this occurs, the measures explained in Section 3.2 were computed on the largest dataset which learning algorithms were able to process and this fact was specified along with the results.

5 Discussion

The aim in this work is to assess the performance of algorithms in terms of scalability and not simply in terms of error like the great majority of papers

Table 3. Performance measures for classification tasks. Notice that N/A stands for *Not Applicable* and those algorithms with FS significantly better (in terms of *Score*) than their version without FS are labeled with a cross.

(a) Connect-4.

Name	Score	Err	AuTE	AuSE	Te5%	Se5%	Eff
GD	4.83	0.38	5.16e1	0.97	1.08e2	1.00e2	0.43
CFS+GD	4.67 [†]	0.51	1.01e1	1.24	1.39e4	1.00e2	0.26
GDX	4.00	0.31	3.71e1	0.92	7.98e1	6.00e4	0.40
CFS+GDX	2.33 [†]	0.32	9.44e0	0.89	1.70e1	1.00e4	0.25
SCG	4.00	0.21	7.01e1	0.77	2.62e2	1.00e4	0.50
CFS+SCG	2.33 [†]	0.29	9.97e0	0.82	2.28e1	1.00e4	0.31
LM*	4.67	0.23	3.79e2	0.77	7.80e2	1.00e4	0.77
CFS+LM	3.87	0.31	4.79e1	0.87	6.68e1	1.00e4	0.44

* Largest training set it can deal with: 1e4 samples.

(b) Covertypes.

Name	Score	Err	AuTE	AuSE	Te5%	Se5%	Eff
GD	4.50	0.38	1.24e2	1.20	2.78e2	1.00e3	0.49
CFS+GD	3.50 [†]	0.45	2.74e1	1.36	3.34e1	1.00e2	0.35
GDX	4.50	0.42	4.74e1	1.32	1.01e2	1.00e4	0.41
CFS+GDX	3.33 [†]	0.51	6.81e0	1.41	0.43e0	1.00e2	0.24
SCG	4.17	0.20	1.64e2	0.81	5.80e2	1.00e5	0.55
CFS+SCG	2.67 [†]	0.29	3.51e1	0.86	5.97e1	1.00e3	0.40
LM*	5.33	0.24	6.41e2	0.94	1.74e3	1.00e4	0.84
CFS+LM	5.00	0.32	2.99e2	0.95	5.15e2	1.00e3	0.58

* Largest training set it can deal with: 1e4 samples.

(c) KDD Cup 99.

Name	Score	Err	AuTE	AuSE	Te5%	Se5%	Eff
GD**	4.00	0.13	4.29e1	0.43	5.53e1	1.00e2	0.50
CFS+GD	3.33 [†]	0.16	8.67e0	0.54	5.41e0	1.00e2	0.34
GDX**	4.17	0.15	2.55e1	0.46	5.93e1	1.00e3	0.44
CFS+GDX	1.67 [†]	0.11	4.61e0	0.37	2.15e1	1.00e3	0.30
SCG**	5.83	0.14	1.10e2	0.51	3.54e2	1.00e4	0.55
CFS+SCG	2.33 [†]	0.08	1.13e1	0.31	4.40e1	1.00e4	0.38
LM*	5.50	0.11	2.21e2	0.46	1.24e3	1.00e4	0.80
CFS+LM	4.17 [†]	0.12	3.38e1	0.49	1.47e2	1.00e2	0.46

* Largest training set it can deal with: 1e4 samples. ** 1e5 samples

(d) MNIST.

Name	Score	Err	AuTE	AuSE	Te5%	Se5%	Eff
GD*	4.17	0.36	1.41e2	0.85	2.26e2	1.00e2	0.65
CFS+GD	3.17	0.26	4.04e1	0.69	1.07e2	1.00e3	0.43
GDX*	4.33	0.22	2.30e2	0.66	6.91e2	1.00e3	0.72
CFS+GDX	2.17 [†]	0.21	3.32e1	0.66	9.98e1	1.00e3	0.42
SCG*	4.17	0.05	2.85e2	0.40	1.62e3	1.00e4	0.81
CFS+SCG	3.00 [†]	0.11	4.77e1	0.49	2.42e2	6.00e4	0.50
LM	N/A	N/A	N/A	N/A	N/A	N/A	N/A
CFS+LM	4.67 [†]	0.13	3.22e2	0.56	1.10e3	1.00e4	0.80

* Largest training set it can deal with: 1e4 samples.

in the literature. The six scalar measures defined in Section 3.2 are considered to evaluate the scalability of learning algorithms, trying to achieve a balance among them. It was expected that some measures were negatively affected by the dimensionality reduction (such as *AuSE*), whilst other measures were positively affected (*AuTE*), since with a dimensionality reduction the algorithms could deal with larger samples employing the same execution time.

In general, the previous results without FS show a lower error at the expense of a longer training time and the results after applying FS present a shorter training time at the expense of a scarcely higher error (in 5 cases the error maintains or improves while in the remaining 11 cases it increases in a range from 1% to 13%). Regarding *AuSE*, even when FS performs slightly worse in 9 out of 16 cases, notice that *Se5%* is maintained or improved. Consequently, we can assert that the FS process has not removed important information of the data.

Since the assessment of the scalability of learning algorithms is a multi-objective problem and there is no possibility of defining a single optimal order of importance among measures, we have opted to focus on the general measure of scalability (*Score*). With regard to this score, Table 3 shows that applying feature selection improves in all cases the previous results. Moreover, the best score for all datasets was obtained after applying the CFS filter (highlighted in bold face) and the distance between the best *Score* with and without applying feature selection is 1.88 in average for all 4 datasets tested in favor of the first approach.

Regarding the different machine learning algorithms tested and observing the results shown in Table 3, feature selection has a small influence on the scalability of GD whereas leads to a great improvement on the scalability of GD_X and SCG algorithms (up to 3.50 according to the *Score* measure for SCG on the KDD Cup 99 dataset). Specially important is the improvement in LM, due to the fact that now it is able to train on MNIST dataset. Plus, all the algorithms are now able to train on all available samples, as the spatial complexity was reduced.

In light of the above, the benefits of the FS process on the scalability of ANNs seems to be apparently demonstrated, moreover with a huge reduction in the number of features required (see Table 2).

6 Conclusions

With the advent of high dimensionality problems, machine learning researchers are not focused only on accuracy but also on scalability. Therefore, sometimes a degradation in the quality of learning is permitted at the expense of turning an impractical algorithm into a practical one, where the crucial issue becomes now how large a problem you can deal with. FS can be helpful in this scenario since it aims at reducing the input dimensionality. In this work, the influence of FS on the scalability of several training algorithms for ANNs were tested, using the well-known CFS filter. Since there are no standard measures of scalability, those defined in the *PASCAL Large Scale Learning Challenge* were used to assess the

scalability of algorithms in terms of error, computational effort, allocated memory and training time. Results show that applying FS improves previous results where no FS was executed, even allowing certain algorithms to be able to train on some datasets in cases where it was impossible due to the spatial complexity. As future work, we plan to extend the study checking the performance of different feature selection methods.

References

1. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.: Feature Extraction. Foundations and Applications. Springer, Heidelberg (2006)
2. Bolon-Canedo, V., Sanchez-Marono, N., Alonso-Betanzos, A.: On the Effectiveness of Discretization on Gene Selection of Microarray Data. In: Proceedings of the International Joint Conference on Neural Networks, pp. 3167–3174 (2010)
3. Bolon-Canedo, V., Sanchez-Marono, N., Alonso-Betanzos, A.: Feature Selection and Classification in Multiple Class Datasets: An Application to KDD Cup 99 Dataset. Journal of Expert Systems with Applications (38), 5947–5957 (2011)
4. Dong, J.: Speed and accuracy: large-scale machine learning algorithms and their applications. Concordia University Montreal, PQ (2003)
5. Sonnenburg, S., Ratsch, G., Rieck, K.: Large scale learning with string kernels. Journal of Large-Scale Kernel Machines, 73–104 (2007)
6. Bottou, L., Bousquet, O.: The tradeoffs of large scale learning. Journal of Advances in Neural Information Processing Systems 20, 161–168 (2008)
7. Catlett, J.: Megainduction: machine learning on very large databases. Ph.D. dissertation, School of Computer Science, University of Technology, Sydney, Australia (1991)
8. Provost, F., Kolluri, V.: A survey of methods for scaling up inductive algorithms. Journal of Data Mining and Knowledge Discovery 3(2), 131–169 (1999)
9. Peteiro-Barral, D., Guijarro-Berdinas, B., Pérez-Sánchez, B., Fontenla-Romero, O.: On the Scalability of Machine Learning Algorithms for Artificial Neural Networks. Journal of IEEE Transactions on Neural Networks (under review)
10. Sonnemburg, S., Franc, V., Yom-Tov, E., Sebag, M.: PASCAL Large Scale Learning Challenge. Journal of Machine Learning Research (2009)
11. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research 5, 1205–1224 (2004)
12. Hall, M.A.: Correlation-based Feature Selection for Machine Learning. PhD thesis, University of Waikato, Hamilton, New Zealand (1999)
13. Zhao, Z., Liu, H.: Searching for Interacting Features. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1156–1167 (1991)
14. Dash, M., Liu, H.: Consistency-based Search in Feature Selection. Journal of Artificial Intelligence 151(1-2), 155–176 (2003)
15. Bishop, C.M.: Pattern recognition and machine learning. Springer, New York (2006)
16. Møller, M.F.: A scaled conjugate gradient algorithm for fast supervised learning. Journal of Neural Networks 6(4), 525–533 (1993)
17. More, J.: The Levenberg-Marquardt algorithm: implementation and theory. Journal of Numerical Analysis, 105–116 (1978)
18. Weiss, S.M., Kulikowski, C.A.: Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems. Morgan Kaufmann, San Francisco (1991)
19. Hecht-Nielsen, R.: Neurocomputing. Addison-Wesley, Menlo Park (1990)

Using Model Trees and Their Ensembles for Imbalanced Data*

Juan J. Rodríguez, José F. Díez-Pastor, César García-Osorio, and Pedro Santos

University of Burgos, Spain

jjrodriguez@ubu.es, jfdiez@beca.ubu.es, cgosorio@ubu.es,

psgonzalez@ubu.es

<http://pisuerga.inf.ubu.es/ADMIRABLE/>

Abstract. Model trees are decision trees with linear regression functions at the leaves. Although originally proposed for regression, they have also been applied successfully in classification problems. This paper studies their performance for imbalanced problems. These trees give better results than standard decision trees (J48, based on C4.5) and decision trees specific for imbalanced data (CCPDT: Class Confidence Proportion Decision Trees). Moreover, different ensemble methods are considered using these trees as base classifiers: Bagging, Random Subspaces, AdaBoost, MultiBoost, LogitBoost and specific methods for imbalanced data: Random Undersampling and SMOTE. Ensembles of Model Trees also give better results than ensembles of the other considered trees.

Keywords: Imbalanced data, Model Trees, Decision Trees, Ensembles.

1 Introduction

In imbalanced datasets, the proportion of the classes is rather different. This situation can require specific methods. Although conventional classification techniques could deal with these data to some extent, it is often possible to have better results using specific methods for these problems. Moreover, even the performance measures could be inadequate. The accuracy or error in an imbalance problem give more importance to the majority class than the minority class. In the classifier construction methods are usually designed for optimizing this measures, it is expected that they will work worse than methods designed specifically for imbalance. One of used measures for imbalance data is the Area Under the Curve (AUC) [1], where the curve refers to the ROC (Receiver Operating Characteristic) curve.

Among the proposed specific methods for imbalanced data, there are decision tree methods [2,3]. However, the AUC is calculated from the confidence assigned by the classifier to the examples. Hence, a method that is able to assign continuous values of the confidence, as are Model Trees, could be more adequate than decision trees that assign the same probabilities to all the examples that end in

* This work was supported by the Project TIN2011-24046 of the Spanish Ministry of Science and Innovation.

the same leaf. One objective of this work is to study the performance of Model Trees, compared with other types of trees, for imbalanced data.

Ensembles [4] are combinations of classifiers, in many situations they give better results than individual classifiers. They have been used successfully for dealing with imbalanced data [5,6,7]. The combined classifiers are called base or member classifiers. Decision trees are often used as base classifiers. Another objective of this work is to study the performance of different ensemble methods when using Model Trees as base classifiers, for imbalanced data.

The rest of the paper is organized as follows. Section 2 briefly describe the methods: trees and ensembles. The experiments are presented in Section 3. Finally, Section 4 presents some concluding remarks.

2 Methods

This section describe the tree methods used as base classifiers and the ensemble methods that are used in the experimental study.

2.1 Tree Methods

Decision trees are one of the most common classification methods. They are particularly adequate for ensembles because they are fast and unstable. In ensembles the computation time is multiplied by the number of base classifiers, in order to have a sensible time is necessary to have fast base classifiers. A method for constructing classifiers is unstable if small changes in the dataset can cause important differences in the classifiers. This is useful in ensembles because the diversity of the base classifiers is necessary for successful ensembles.

One of the most well-known and more used methods for Decision Trees is C4.5 [8]. In these trees, in the internal nodes one of the attributes is considered and a child is selected depending on the values of the attributes. In the leaves, one of the classes is predicted, although it is also possible to obtain a probability for each class.

There have been some proposals for decision tree methods specific for imbalanced data [2,3]. In this work, Class Confidence Proportion Decision Trees (CCPDT) [3] are used. They are inspired by C4.5, but using variants of the Entropy and Information Gain. The variants are based on the Class Confidence Proportion (CCP) measure. Given a rule $X \rightarrow y$, it is defined as

$$CCP(X \rightarrow y) = \frac{tpr}{tpr + fpr}$$

Where tpr is the True Positive Rate and fpr is the False Positive Rate.

From the CCP measure, a variant of the Entropy is defined:

$$Entropy^{CCP}(t) = - \sum_j CCP(X \rightarrow y_j) \log CCP(X \rightarrow y_j)$$

Using this Entropy, a variant of the Information Gain is obtained to be used in CCPDT. These trees are described in detail in [3].

Model trees [9,10] are decision trees that have linear regression functions at the leaves. Initially, they were proposed for regression, but in [11] they were used for classification. For each class a Model Tree is constructed, the labels are 1 or 0 for the examples of the corresponding class and the examples of the other classes, respectively.

2.2 Ensemble Methods

In this paper, we are considering homogeneous ensembles, that is, the combined classifiers are obtained using the same method. In order to obtain different classifiers using the same method, a usual approach is to train the classifiers with modified datasets.

In Bagging [12], each base classifier is trained using a random sample, with replacement, of the training data. Some examples of the training will appear several times in the sample, while others will not appear.

In Random Subspaces [13], all the training data is used to train all the base classifiers, but the classifiers are trained in different random subspaces.

AdaBoost [14] is based on assigning a weight to each training example. When a base classifier is constructed the examples weights are modified increasing them for misclassified examples. These weights have to be taken into account when constructing the next classifier. An easy way is to train the classifiers using a weighted sample (according to the example weights) from the training data.

MultiBoost [15] is a variant of Adaboost, the difference is that when a certain number of classifiers (called sub-committee) have been constructed, the weights are randomly initialized (following a distribution that is based on the number of times an example would be selected for a sample in Bagging) and then the process continues as in Adaboost until the number of classifiers constitutes again a sub-committee or enough classifiers had been generated.

LogitBoost [16] is another method inspired by AdaBoost and based on Logistic Regression. It also combines several models, but this models are not classifiers but regressors, because their task is to predict a numeric value instead of a nominal category. LogitBoost can be used directly with Model Trees, since they predict numeric values. For the other ensemble methods, Model Trees are used but transforming the classification problem in a regression problem.

Random Undersampling [17,6] and SMOTE [18] are two techniques for dealing with imbalanced data. They generate a more balanced dataset, a classifier is constructed using this dataset instead of the original one. They can be used to construct ensembles, each member classifier is constructed from a different generated dataset. These datasets are different because in their generation random values are used.

In Random Undersampling all the examples of the minority class appear in the generated dataset, while for the majority class a random sample is selected. The sample size is smaller than the number of examples in the majority class making the generated dataset more balanced.

In SMOTE, the generated data has all the examples in the original data but additional synthetic examples for the minority class are also included. The

Table 1. Characteristics of the datasets

Dataset	Examples	Attributes		Minority percentage	References
		Numeric	Nominal		
adult	48842	6	8	23.93	[19]
breast-w	699	9	0	34.48	[19,7]
breast-y	286	0	9	29.72	[19,7]
credit-g	1000	7	13	30.00	[19,7]
ecg ¹	200	304	0	33.50	[20]
fourclass ²	862	2	0	35.61	[3,7]
haberman	306	3	0	26.47	[19]
heart-s	123	5	8	6.50	[19]
heart-v	200	5	8	25.50	[19,7]
hypo	3163	7	18	4.77	[19,7]
laryngeal2 ³	692	16	0	7.66	[21]
musk-2	6598	166	0	15.41	[19]
phoneme ⁴	5404	5	0	29.35	[3,7]
pima	768	8	0	34.90	[19,3,7]
sick	3772	7	22	6.12	[19]
svmguide1 ²	3089	4	0	35.25	[22,3,7]
tic-tac-toe	958	0	9	34.66	[19,7]
wafer ¹	1194	1188	0	10.64	[20]

1: <http://www.cs.cmu.edu/~bobski/pubs/tr01108.html>

2: <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

3: http://www.bangor.ac.uk/~mas00a/activities/real_data.htm

4: <http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/databases/REAL/phoneme/>

process for generating a synthetic example is to take an example of the minority class and then to select randomly one of its nearest neighbours (the number of neighbours is a parameter of the algorithm). The new example is generated by randomly choosing a point in the line that connects the example and the selected neighbour.

3 Experiments

3.1 Datasets

Table 1 shows the datasets used in the study. All of them are two-classes datasets. Most of them are from the UCI repository [19]. The source for the rest is shown in the table.

3.2 Settings

The experiments were performed using Weka [23]. The results were obtained using 5×2 folds stratified cross validation [24]. Two performance measures were considered: accuracy and the area under the ROC curve (AUC). The second is more adequate for imbalanced data. Unless explicitly specified, the parameters for the different methods take the default values given by Weka.

Three types of trees were considered as base classifiers: model trees (M5P), standard trees (J48) and a specific method for imbalance data (CCPDT). For the three methods, both pruned and unpruned trees were considered. They are denoted with (P) and (U), respectively. For J48 and CCPDT, Laplace smoothing [3] was used on the leaves.

Ensemble size was 100. For MultiBoost, the size of the sub-committees was 10. For the Random Subspaces method, two subspace sizes were considered: 50% and 75% of the original space size.

In the case of Undersampling, the sample size is the number of instances in the minority class. For SMOTE, the number of examples in the minority class is doubled by adding as many artificial examples as original examples in the minority class.

LogitBoost was used only with M5P, because it combines regression models, and J48 and CCPDT generate classification models.

The total number of configurations is 56: six tree configurations, eight ensemble methods combined with the six base classifiers and LogitBoost with only two tree configurations.

3.3 Results

Average ranks [25] are used for comparing the methods using the considered datasets. For each dataset, the methods are ordered from best to worst. The best method is given rank 1, the second rank 2, and so on. If some methods have the same results, an average value is assigned to them (e.g, if 4 methods have the best result, all of them have a rank of 2.5). The average rank for a method is the average value across all the datasets.

Fig. 1 shows the average ranks using the AUC. These average ranks are for the six considered base methods (three tree types, with or without pruning). Each considered ensemble method is constructed using these base classifiers, the average ranks are obtained from these six ensembles. The figure also shows the average ranks when using the base methods as single classifiers.

For all the ensemble methods, with only one exception (Random Subspaces 75%), the best average rank is for one of the two M5P configurations.

Fig. 2 also shows the average ranks, but using the AUC. In this case, for all the ensemble methods the best average rank is for one of the two M5P configurations. Moreover, with only one exception (AdaBoost), the two M5P configurations have the two best average ranks.

Iman and Davenport test [26,25] checks whether the measured average ranks are significantly different from the mean rank, 3.5 for six datasets. Table 2 shows the p -values for this test. For a single classifier the differences in the average ranks are very significant, but for the majority of the ensemble methods there is not a significant difference (for a significance level of $\alpha = 0.05$).

When there is a significant difference according to the Iman and Davenport test, we can proceed with post-hoc tests [26,25]. For instance, Table 3 shows the adjusted p -values according to different procedures when using single trees and the AUC as the performance measure.

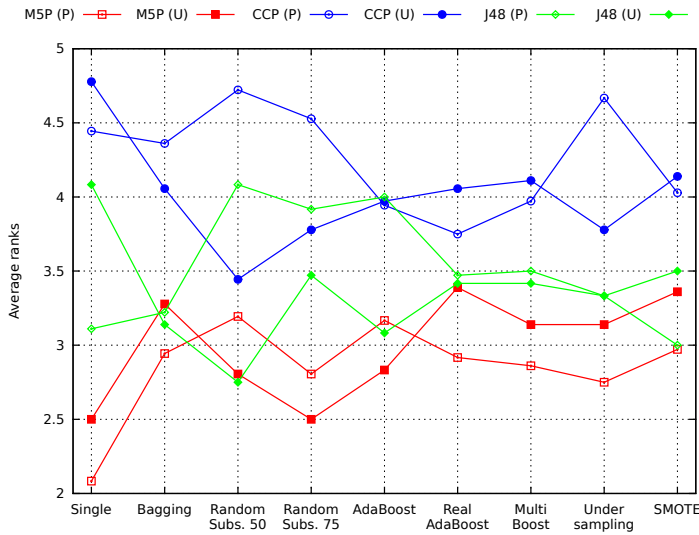


Fig. 1. Average ranks, for each ensemble method, using the accuracy as the performance measure

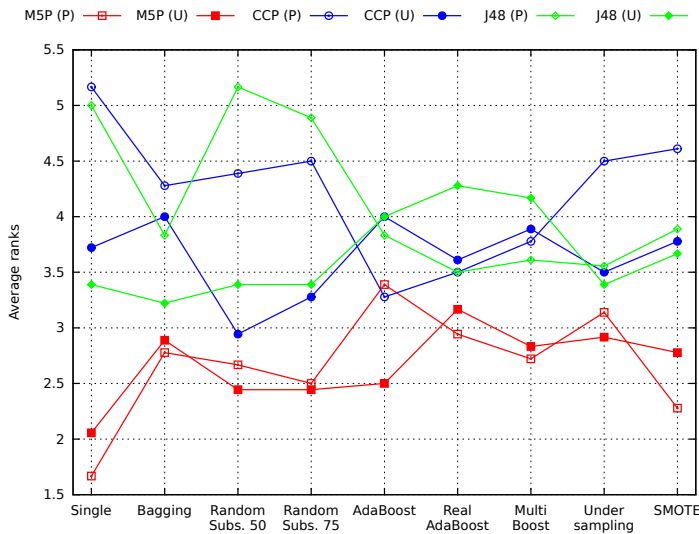


Fig. 2. Average ranks, for each ensemble method, using the AUC as the performance measure

Fig. 3 also shows the average ranks for the accuracy, but in this case considering all the configurations (combinations of ensemble methods and base classifiers). The figure also shows the average value for an ensemble method across the different base classifiers, and the average values for a base classifier across the ensemble methods. The rows (ensemble methods) are sorted according to the average of the values in the row. According to this average, the best ensemble

Table 2. P-values computed by Iman and Davenport Test

Ensemble	Accuracy	AUC
Single	● 7.665509918087394E-7	● 1.0147538473137857E-15
Bagging	0.1366314640196765	0.07106793559879934
Subspaces 50	● 0.006528416273527019	● 2.328839036947886E-6
Subspaces 75	● 0.010285938821396369	● 1.6032655507007264E-5
AdaBoost	0.20554522420334784	0.12214516266073304
Real AdaBoost	0.5944526335263839	0.3823273345870149
MultiBoost	0.3225087277934808	0.10800966754539121
Undersampling	● 0.04162332232208745	0.17510955274240894
SMOTE	0.2757036904317331	● 0.001667049907520199

Values smaller than 0.05 are marked with the symbol “●”.

Table 3. Adjusted p -values obtained for single trees, according to the AUC and using different procedures

	hypothesis	unadjusted	Nememyi	Holm	Shaffer	Bergmann
1	M5P (P) vs CCPDT (P)	1.994403e-08	2.991604e-07	2.991604e-07	2.991604e-07	2.991604e-07
2	M5P (P) vs J48 (P)	9.030489e-08	1.354573e-06	1.264268e-06	9.030489e-07	9.030489e-07
3	M5P (U) vs CCPDT (P)	6.073145e-07	9.109717e-06	7.895088e-06	6.073145e-06	6.073145e-06
4	M5P (U) vs J48 (P)	2.339790e-06	3.509684e-05	2.807747e-05	2.339790e-05	1.403874e-05
5	M5P (P) vs CCPDT (U)	9.799455e-04	1.469918e-02	1.077940e-02	9.799455e-03	6.859619e-03
6	CCPDT (P) vs J48 (U)	4.361123e-03	6.541685e-02	4.361123e-02	4.361123e-02	3.052786e-02
7	M5P (P) vs J48 (U)	5.750137e-03	8.625206e-02	5.175124e-02	4.361123e-02	3.450082e-02
8	M5P (U) vs CCPDT (U)	7.526315e-03	1.128947e-01	6.021052e-02	5.268421e-02	3.450082e-02
9	J48 (P) vs J48 (U)	9.779628e-03	1.466944e-01	6.845740e-02	6.845740e-02	3.911851e-02
10	CCPDT (P) vs CCPDT (U)	2.054385e-02	3.081578e-01	1.232631e-01	1.232631e-01	8.217541e-02
11	M5P (U) vs J48 (U)	3.250944e-02	4.876417e-01	1.625472e-01	1.300378e-01	8.217541e-02
12	CCPDT (U) vs J48 (P)	4.046184e-02	6.069275e-01	1.625472e-01	1.618473e-01	8.217541e-02
13	M5P (P) vs M5P (U)	5.328840e-01	7.993260e+00	1.598652e+00	1.598652e+00	1.598652e+00
14	CCPDT (U) vs J48 (U)	5.929801e-01	8.894701e+00	1.598652e+00	1.598652e+00	1.598652e+00
15	CCPDT (P) vs J48 (P)	7.892680e-01	1.183902e+01	1.598652e+00	1.598652e+00	1.598652e+00

method is MultiBoost and the worst is Undersampling. The configuration that is in the top rank is MultiBoost with M5 (P).

Fig. 4 shows the average ranks of the AUC for all the configurations. In this case the best ensemble method is Bagging and the best configuration is Bagging with M5P (P). Undersampling is the second best ensemble method, while for accuracy it was the worst.

For these figures, if we consider the average values of the ensemble methods across the six base classifiers and the average of the base classifiers across all the ensemble methods, it can be observed that the differences among ensemble methods are much more relevant than among the base classifiers.

For the average ranks from all the configurations, the p -values computed by the Iman and Davenport test are 1.068817e-47 for the accuracy and 1.110223e-16 for the AUC, respectively. Post-hoc tests are not useful for so many configurations, 56, because there are too many pairwise comparisons. For instance, according to the Nemenyi test with a confidence level of 0.05, two methods are significantly different if the average ranks differ in at least 21.988.

MultiBoost	15.278	17.333	22.000	22.583	18.222	17.528	18.824
Bagging	17.889	18.167	25.806	23.917	19.306	20.194	20.880
AdaBoost	19.528	19.333	23.278	22.806	21.972	20.194	21.185
LogitBoost	21.972	25.75	X	X	X	X	23.861
Real AdaBoost	23.528	25.111	23.306	25.528	23.222	23.722	24.069
Subspaces 75	21.222	20.833	33.028	29.528	27.361	26.472	26.407
Subspaces 50	27.500	25.583	34.722	28.028	30.778	27.694	29.051
SMOTE	30.028	31.5	38.111	39.750	30.389	34.056	33.972
Single	28.111	30.694	42.278	44.639	34.944	40.944	36.935
Undersampling	43.167	44.806	50.611	47.194	47.167	47.389	46.722
AVERAGE	24.822	25.911	32.571	31.552	28.151	28.688	

M5 (P) M5 (U) CCP (P) CCP (U) J48 (P) J48 (U) AVER.

Fig. 3. Average ranks, for all the considered configurations, using the accuracy as the performance measure (lighter color means better)

Bagging	11.611	12.500	24.833	23.778	17.556	16.000	17.713
Undersampling	16.306	16.583	29.528	24.722	20.222	19.889	21.208
Subspaces 50	15.139	15.083	26.694	19.694	31.222	21.139	21.495
LogitBoost	21.889	23.778	X	X	X	X	22.833
Subspaces 75	14.583	15.861	30.361	24.694	33.528	24.667	23.949
SMOTE	17.611	19.778	32.722	29.944	25.722	25.833	25.269
Real AdaBoost	30.389	32.278	32.556	33.222	33.167	35.778	32.898
MultiBoost	34.639	34.500	39.028	39.917	37.583	40.583	37.708
AdaBoost	37.194	34.889	37.472	39.417	41.083	40.806	38.477
Single	24.111	28.333	50.667	44.111	48.278	42.528	39.671
AVERAGE	22.347	23.358	33.762	31.056	32.040	29.691	

M5 (P) M5 (U) CCP (P) CCP (U) J48 (P) J48 (U) AVER.

Fig. 4. Average ranks, for all the considered configurations, using the AUC as the performance measure (lighter color means better)

4 Conclusions

For imbalance datasets, model trees (M5P) are significantly better than standard decision trees (J48) and decision trees designed specifically for dealing with imbalance (CCPDT). Moreover, for the considered datasets, CCPDT is not an improvement over J48.

When using ensembles of trees, model trees are still better (have better average ranks) than the other considered trees. Although, in this case, the differences are smaller and generally are not significant. Nevertheless, for almost all the considered ensemble methods, the top position according to the average rank is for ensembles of model trees.

For the performance of the considered classifier configurations, the ensemble method is more relevant than the type of tree used as base classifier.

Among the ensemble methods used in the experiments, according to the accuracy, Multiboost is the best, but if the AUC is considered, the best method is Bagging. Although these methods are not specifically designed for imbalance problems, they give better results than Undersampling and SMOTE.

It is possible to combine several of the considered ensemble strategies. For instance, Boosting and SMOTE [5], Boosting and Undersampling [6], Random Subspaces and SMOTE or Undersampling [7]. For future work, these and other combinations can be used with Model Trees as base classifiers.

Recently, it has been argued that AUC is not coherent [27], although there are also coherent interpretations of this measure [28]. The experimental study presented in this paper can be extended using other measures, such as the H measure from [27] or the area under the cost curve [28].

Acknowledgements. We wish to thank the developers of Weka [23], CCPDT [3] and the statistical tests [26]. We also express our gratitude to the donors of the different datasets and the maintainers of the UCI Repository [19].

References

1. Fawcett, T.: An introduction to ROC analysis. *Pattern Recognition Letters* 27, 861–874 (2006)
2. Cieslak, D., Chawla, N.: Learning decision trees for unbalanced data. In: Daelemans, W., Goethals, B., Morik, K. (eds.) *ECML PKDD 2008, Part I*. LNCS (LNAI), vol. 5211, pp. 241–256. Springer, Heidelberg (2008)
3. Liu, W., Chawla, S., Cieslak, D.A., Chawla, N.V.: A Robust Decision Tree Algorithm for Imbalanced Data Sets. In: *10th SIAM International Conference on Data Mining, SDM 2010*, pp. 766–777. SIAM (2010)
4. Kuncheva, L.I.: *Combining Pattern Classifiers: Methods and Algorithms*. Wiley Interscience (2004)
5. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: Improving prediction of the minority class in boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) *PKDD 2003*. LNCS (LNAI), vol. 2838, pp. 107–119. Springer, Heidelberg (2003)
6. Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 40, 185–197 (2010)
7. Hoens, T., Chawla, N.: Generating Diverse Ensembles to Counter the Problem of Class Imbalance. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) *PAKDD 2010*. LNCS, vol. 6119, pp. 488–499. Springer, Heidelberg (2010)

8. Quinlan, J.R.: C4.5: Programs for Machine Learning. Machine Learning. Morgan Kaufmann, San Mateo (1993)
9. Quinlan, R.J.: Learning with continuous classes. In: 5th Australian Joint Conference on Artificial Intelligence, pp. 343–348. World Scientific, Singapore (1992)
10. Wang, Y., Witten, I.H.: Induction of model trees for predicting continuous classes. In: van Someren, M., Widmer, G. (eds.) ECML 1997. LNCS, vol. 1224, Springer, Heidelberg (1997)
11. Frank, E., Wang, Y., Inglis, S., Holmes, G., Witten, I.H.: Using model trees for classification. *Machine Learning* 32, 63–76 (1998)
12. Breiman, L.: Bagging predictors. *Machine Learning* 24, 123–140 (1996)
13. Ho, T.K.: The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 832–844 (1998)
14. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55, 119–139 (1997)
15. Webb, G.I.: Multiboosting: A technique for combining boosting and wagging. *Machine Learning* 40(2), 159–196 (2000)
16. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. *Annals of Statistics* 95, 337–407 (2000)
17. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39, 539–550 (2009)
18. 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)
19. Frank, A., Asuncion, A.: UCI machine learning repository (2010), <http://archive.ics.uci.edu/ml>
20. Olszewski, R.T.: Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data. PhD thesis, Computer Science Department, Carnegie Mellon University (2001)
21. Kuncheva, L.I., Hadjitodorov, S.T., Todorova, L.P.: Experimental comparison of cluster ensemble methods. In: FUSION 2006, Florence, Italy (2006)
22. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University (2003)
23. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. *SIGKDD Explorations* 11 (2009)
24. Dietterich, T.G.: Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation* 10, 1895–1923 (1998)
25. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
26. García, S., Herrera, F.: An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research* 9, 2677–2694 (2008)
27. Hand, D.: Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning* 77, 103–123 (2009)
28. Flach, P., Hernandez-Orallo, J., Ferri, C.: A coherent interpretation of auc as a measure of aggregated classification performance. In: 28th International Conference on Machine Learning (ICML 2011), pp. 657–664. ACM (2011)

Instance Selection for Class Imbalanced Problems by Means of Selecting Instances More than Once^{*}

Javier Pérez-Rodríguez, Aida de Haro-García, and Nicolás García-Pedrajas

Department of Computing and Numerical Analysis, University of Córdoba, Spain
javier@cibr.org, {adeharo,npedrajas}@uco.es
<http://www.cibr.org/>

Abstract. Although many more complex learning algorithms exist, k -nearest neighbor (k -NN) is still one of the most successful classifiers in real-world applications. One of the ways of scaling up the k -nearest neighbors classifier to deal with huge datasets is instance selection. Due to the constantly growing amount of data in almost any pattern recognition task, we need more efficient instance selection algorithms, which must achieve larger reductions while maintaining the accuracy of the selected subset.

However, most instance selection methods do not work well in class imbalanced problems. Most algorithms tend to remove too many instances from the minority class. In this paper we present a way to improve instance selection for class imbalanced problems by allowing the algorithms to select instances more than once. In this way, the fewer instances of the minority can cover more portions of the space, and the same testing error of the standard approach can be obtained faster and with fewer instances. No other constraint is imposed on the instance selection method.

An extensive comparison using 40 datasets from the UCI Machine Learning Repository shows the usefulness of our approach compared with the established method of evolutionary instance selection. Our method is able to, in the worst case, match the error obtained by standard instance selection with a larger reduction and shorter execution time.

1 Introduction

Although many more complex learning algorithms exist, k -nearest neighbor (k -NN) is still one of the most successful classifiers in real-world applications [12] [25]. However, the overwhelming amount of data available nowadays in any field of research poses new problems for classification algorithms. This huge amount of data makes most of the existing algorithms inapplicable to many real-world problems due to scalability issues, and k -NN is not an exception. Two approaches

^{*} This work was supported in part by the Project TIN2008-03151 of the Spanish Ministry of Science and Innovation and the project P09-TIC-4623 of the Junta de Andalucía.

have been used to face this problem: fast nearest neighbor calculation [3] [27] and instance selection [4]. Unfortunately, fast neighbor searches may be difficult to implement and usually require approximations that damage performance. On the other hand, instance selection obtains smaller subsets that can be efficiently searched for neighbors. Additionally, instance selection can also be applied to other instance-based classifiers, and other problems, such as multiple-instance learning [14].

Instance selection [22] consists of choosing a subset of the total available data to achieve the original purpose of the data mining application as if the whole data were used. We can distinguish two main models [6]: instance selection as a method for prototype selection for algorithms based on prototypes (such as k -nearest neighbors) and instance selection to obtain the training set, for a learning algorithm that uses a training set (such as classification trees or neural networks).

The problem of instance selection for instance based learning can be defined as [4] “the isolation of the smallest set of instances that enable us to predict the class of a query instance with the same (or higher) accuracy than the original set”.

Cano et al. [6] performed a comprehensive comparison of the performance of different algorithms for instance selection. They compared a generational genetic algorithm [20], a steady-state genetic algorithm [29], a CHC genetic algorithm [10], and a population based incremental learning algorithm [1] along with most of the non-evolutionary algorithms. They found that evolutionary based methods were able to outperform classical algorithms in both classification accuracy and data reduction. Among the evolutionary algorithms, CHC was able to achieve the best overall performance.

It has been shown that different groups of learning algorithms need different instance selectors to suit their learning/search bias [5]. This may render many instance selection algorithms useless, if their philosophy of design is not suitable for the problem at hand.

It has been repeatedly shown that most classification methods suffer from an imbalanced distribution [2] [23] of the training instances among the classes [7]. Most learning algorithms expect an approximately even distribution of the instances among the different classes and suffer, in different degrees, when that is not the case. Dealing with the class imbalanced problem is a difficult task, but a very relevant one as many of the most interesting and challenging real-world problems have a very uneven class distribution, such as gene recognition, intrusion detection, web mining, etc.

In most cases this problem appears in two class datasets. There is a class of interest, the positive class, which is highly underrepresented in the dataset, together with a negative class which accounts for most of the instances. In highly imbalanced problems the ratio between the positive and the negative class can be as high as 1:1000 or 1:10000. Many algorithms and methods have been proposed to ameliorate the effect of class imbalanced on the performance of the learning algorithms. There are mainly three different approaches [28] [15]:

- Internal approaches acting on the algorithm. These approaches modify the learning algorithm to deal with the imbalance problem. They can adapt the

decision threshold to create a bias towards the minority class or introduce costs in the learning process to compensate the minority class.

- External approaches acting on the data. These algorithms act on the data instead of on the learning method. They have the advantage of being independent from the classifier used. There are two basic approaches, oversampling the minority class and undersampling the majority class.
- Combined approaches which are based on *boosting* [13] taking into account the imbalance in the training set. These methods modify the basic boosting method to account for the minority class underrepresentation in the dataset.

Instance selection methods, when applied in class imbalanced cases, tend to remove too many instances of the minority class, damaging their performance [17]. In this paper we present simple way to improve instance selection for the class imbalanced case through an approach that is easy to use. Standard instance selection, both classical and evolutionary, selects each instance once or not at all. We propose selecting instances more than once. In this way, instead of choosing whether an instance is selected or not, we have all the possible choices of selection, from 0 to $k/2$, k being the number of nearest neighbors to consider.

The underlying idea is to cover the same amount of space with fewer instances. Figure 1 illustrates this hypothesis. We have two close instances, x_1 and x_2 , of the same class, and a query instance, q , of the same class. Using 3 nearest neighbors we need to select both instances, x_1 and x_2 , to correctly classify the query instance. Otherwise, instances y_1 and y_2 will be nearer to query pattern q , and the classification will be wrong. However, if we select x_1 or x_2 twice, the classification will be correct with only one instance. The selection of instances more than once can be stored using a few bits depending on the value of k , and the storage reduction achieved can be improved.

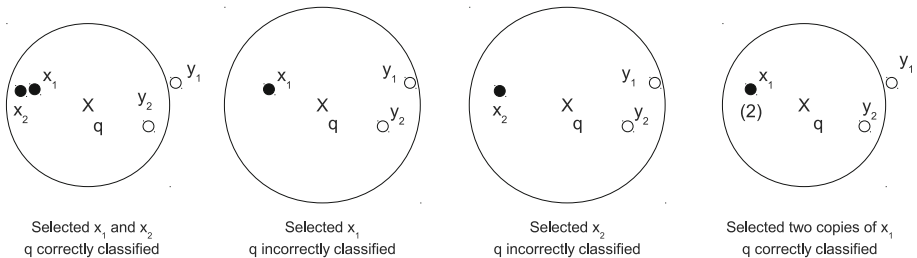


Fig. 1. An example of the usefulness of selecting instances more than once with $k = 3$

Furthermore, as fewer distances must be calculated, instance selection will be faster. The reduction in execution time will be more marked for larger values of k .

This paper is organized as follows: Section 2 presents the proposed model for multi-selection of instances; Section 3 describes the experimental setup; Section 4 describes the results of the experiments; and Section 5 provides the conclusions of our work and our suggestions for future lines of research.

2 Multiple-Selection of Instances

As we have stated, our approach consists of allowing the selection of instances by any algorithm of our choice more than once. For a value of k neighbors, an instance can be selected from one to $k/2$ times. Selecting an instance more than $k/2$ times is useless, as once it is included in the neighborhood of a query instance, $k/2$ copies are enough to decide the classification.

This idea is general enough to be used with any instance selection algorithm. Cano et al. [6] performed a thorough comparison of most instance selection algorithms, both classical and evolutionary, and found CHC to be the overall best of both evolutionary and non-evolutionary algorithms. Thus, in our experiments, we will compare our approach using a CHC genetic algorithm as instance selection algorithm. CHC [10] stands for *Cross generational elitist selection, Heterogeneous recombination and Cataclysmic mutation*. We have made use of the CHC genetic algorithm in its standard form [24].

The implementation of the genetic algorithm uses the most natural representation for each individual. The chromosome of each individual has as many bits as there are instances in the training set. A bit whose value is 1 means that the corresponding instance is selected, and a 0 means that the corresponding instance is not selected. The fitness measure of an individual i , f_i is given by:

$$f_i = w \cdot \text{acc}_i + (1 - w) \cdot \rho_i, \quad (1)$$

where ρ_i is the reduction achieved by the individual, acc_i is the accuracy, measured using an appropriate class imbalanced value, and $w = 1/2$, giving the same weight for both objectives. Obviously, the objective of the genetic algorithm is to maximize accuracy and to minimize storage requirements. The relative weights of storage and accuracy are selected to avoid a large reduction with the side effect of a poor performance. However, although we have opted for the CHC algorithm, any instance selection algorithm can be adapted to work with multi-selection of instances.

The standard version of CHC algorithm must be slightly modified to account for the multi-selection of instances. In the initialization of the population, we use a probability p_s of selecting a certain instance. For multi-selection, this probability is used to decide whether to select an instance. In case the instance is selected, the number of times it is selected is chosen in the interval $[1, k/2]$, using a uniform distribution.

We apply the HUX operator without modification. The mutation operator is slightly modified. In the standard case, we apply this operator to an individual with a probability p_m , and then each bit is modified with a probability p_{bit} . For the case of multi-selection, if we have to modify a bit the action performed depends on the value of the bit. If the instance is selected, no matter how many times, it is removed. If the instance is not selected, it is selected a number of times randomly chosen in the interval $[1, k/2]$.

It is known that genetic algorithms usually encounter problems in fine tuning the solution. To avoid this problem we have added a new mutation operator in the

form of a local optimization algorithm. This mutation carries out an adaptation of the reduced nearest neighbor (RNN) rule [19] to $k > 1$. This mutation is applied with a probability p_{rnn} .

3 Experimental Setup

To test the ability of our approach in class imbalanced datasets, we have used the problems shown in Table 1. All problems are two-class problems, with an imbalance ratio of the minority class to the majority class from 1:2 to 1:130. The datasets breast-cancer, cancer, euthyroid, german, haberman, hepatitis, ionosphere, ozone1hr, ozone8hr, phoneme, pima, sick, tic-tac-toe and titanic are from the UCI Machine Learning Repository [11]. The remaining datasets were artificially created following [18]. To estimate the storage reduction and generalization error, we used a 10-fold cross-validation method.

We have used the Wilcoxon test as main statistical test for comparing pairs of algorithms. This test was chosen because it assumes limited commensurability and is safer than parametric tests, as it does not assume normal distributions or homogeneity of variance. Thus, it can be applied to error ratios and storage requirements. Furthermore, empirical results [9] show that it is also stronger than other tests.

When evaluating instance selection algorithms speed considerations are difficult to measure, as we are evaluating not only an algorithm but also a certain implementation. However, as scalability is a common problem in evolutionary instance selection, execution time values are very relevant evaluating the merits of any algorithm. To allow a fair comparison, we performed all the experiments on the same machine, a bi-processor computer with two Intel Xeon QuadCore at 1.60GHz. To perform sound experiments, exactly the same algorithm was used for the standard method and our proposal. That is, when we applied our method and the standard method, the implementation was the same in both cases, with the sole exception of allowing the selection of each instance more than once.

The source code used for all methods as well as the partitions of the datasets, in C and licensed under the GNU General Public License, is freely available from the authors upon request.

4 Experimental Results and Discussion

In this section, we show the experiments we performed aimed at study the behavior of the proposed model in class imbalanced problems.

We hypothesized that selecting instances more than once might implicitly take care of class imbalanced datasets. Thus, evaluating our method will show the validity of this assumption. We must bear in mind that we are not proposing a method for informative undersampling for the class imbalanced problem [15]. Our aim is studying whether our proposal is able to improve the results of the standard CHC algorithm in class imbalanced problems.

Table 1. Summary of datasets with the imbalance ratio (IR) between the minority and majority classes. The features of each dataset can be C(continuous), B(binary) or N(nominal). The Inputs column shows the number of input variables after transforming binary and nominal variables to numerical values.

Data set	Cases	Features			Inputs	IR
		C	B	N		
1 abalone19	4177	7	-	1	10	1:130
2 abalone9-18	731	7	-	1	10	1:17
3 breast-cancer	286	-	3	6	15	1:3
4 cancer	699	9	-	-	9	1:2
5 carG	1728	-	-	6	16	1:25
6 ecoliCP-IM	220	7	-	-	7	1:2
7 ecoliM	336	7	-	-	7	1:4
8 ecoliMU	336	7	-	-	7	1:9
9 ecoliOM	336	7	-	-	7	1:16
10 euthyroid	3163	7	18	-	44	1:10
11 german	1000	6	3	11	61	1:3
12 glassBWFP	214	9	-	-	9	1:3
13 glassBWNFP	214	9	-	-	9	1:2
14 glassContainers	214	9	-	-	9	1:16
15 glassNW	214	9	-	-	9	1:4
16 glassTableware	214	9	-	-	9	1:23
17 glassVWFP	214	9	-	-	9	1:12
18 haberman	306	3	-	-	3	1:3
19 hepatitis	155	6	13	-	19	1:4
20 ionosphere	351	33	1	-	34	1:2
21 new-thyroidT	215	5	-	-	5	1:6
22 optdigitsZ	5620	64	-	-	64	1:10
23 ozone1hr	2536	72	-	-	72	1:34
24 ozone8hr	2534	72	-	-	72	1:15
25 phoneme	5404	5	-	-	5	1:3
26 pima	768	8	-	-	8	1:2
27 satimageF	6435	36	-	-	36	1:10
28 satimageT	6435	36	-	-	36	1:10
29 segmentO	2310	19	-	-	19	1:7
30 sick	3772	7	20	2	33	1:16
31 splice-EI	3175	-	60	120	1:4	
32 Splice-IE	3175	-	60	120	1:4	
33 tic-tac-toe	958	-	9	9	1:2	
34 titanic	2201	-	3	8	1:3	
35 vehicleVAN	846	18	-	-	18	1:4
36 vowelZ	990	10	-	-	10	1:11
37 yeastCYT-POX	483	8	-	-	8	1:24
38 yeastEXC	1484	8	-	-	8	1:42
39 yeastME1	1484	8	-	-	8	1:33
40 yeastME2	1484	8	-	-	8	1:29

For these datasets we performed experiments using three values for k , 3, 7 and 11 neighbors. Both algorithms, standard CHC and our method, were applied with the same parameters. Both versions of the algorithm use a standard CHC algorithm with a population of 100 individuals, evolved for 10,000 generations. The mutation rate is $p_m = 0.1$ and $p_{\text{bit}} = 0.1$, and we applied an additional mutation operator based on the RNN [19] algorithm with a probability of $p_{\text{rnn}} = 0.05$. For the initialization of the population, we set the probability of selecting an instance to $p_s = 0.33$. For the case of our method, when an instance is selected, its initial count was set to a random value in the interval $[1, k/2]$.

Accuracy is not a useful measure for imbalanced data, especially when the number of instances of the minority class is very small compared with the majority class. For an imbalance ratio of 1:100, a classifier that assigns all instances to the majority class will have 99% accuracy. Several measures [28] have been developed to take into account the imbalanced nature of the problems. Given the number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN), we can define the following set of basic measures:

- True positive rate TP_{rate} , recall R or sensitivity. $TP_{rate} = R = S_n = \frac{TP}{TP+FN}$. This measure is relevant if we are only interested in the performance on the positive class.
- True negative rate TN_{rate} or specificity. $TN_{rate} = Sp = \frac{TN}{TN+FP}$.
- False positive rate FP_{rate} . $FP_{rate} = \frac{FP}{TN+FP}$.
- False negative rate FN_{rate} . $FN_{rate} = \frac{FN}{TP+FN}$.
- Positive predictive value PP_{value} or precision P . $PP_{value} = \frac{TP}{TP+FP}$.
- Negative predictive value NP_{value} . $NP_{value} = \frac{TN}{TN+FN}$.

From these basic measures, others have been proposed. The F -measure joins recall and precision in a measure that is a harmonic mean of both, $F = \frac{2RP}{R+P} = \frac{2}{1/R+1/P}$. The harmonic mean of two measures tends to be closer to the smaller one than the arithmetic mean. Thus, F measures if recall and precision both have high values. If we are concerned about performance on both negative and positive classes, the G -mean measure [21] considers both:

$$G - \text{mean} = \sqrt{TP_{rate} \cdot TN_{rate}}. \quad (2)$$

G -mean measures the balance performance of the learning algorithm between the two classes. Among these measures, we have chosen G -mean as the criterion for evaluating the methods. G -mean is used as the measure of accuracy for the fitness function (see eq. 1).

Figure 2 shows the relative movement diagrams for the results for both algorithms. The figure shows results for testing error and storage requirements. This graphic representation is based on the kappa-error relative movement diagrams [26] [16]. Here, however, instead of the kappa difference value, we use the storage difference. The idea of these diagrams is to represent with an arrow the results of two methods applied to the same dataset. The arrow starts at the coordinate origin, and the coordinates of the tip of the arrow are given by the difference between the errors and storage of the standard CHC instance selection algorithm and our method. These graphs are a convenient way of summarizing the results. Arrows pointing up-right represent datasets for which our method outperformed the standard algorithm in both error and storage; arrows pointing up-left indicate that our algorithm improved the storage but had worse testing error; arrows pointing down-right indicate that our algorithm improved the testing error but had a worse storage; and finally, arrows pointing down-left indicate that our algorithm was worse in both testing error and storage.

We can see that the arrows are almost all pointing upwards, which means that our algorithm almost always achieved better storage reduction. In terms of testing error, we found very similar results for both methods. Overall, these results show that our method is also a valid approach for class imbalanced problems.

Table 2 shows the comparison for the three values of k in terms of G -mean measure, storage and execution time. The table shows the win/draw/loss record of our method against the standard approach in the row labeled with an s ; the p -value of a sign test over the win/loss record, labeled p_s ; and the p -value of the Wilcoxon test, labeled p_w . The comparison shows that our proposal achieves

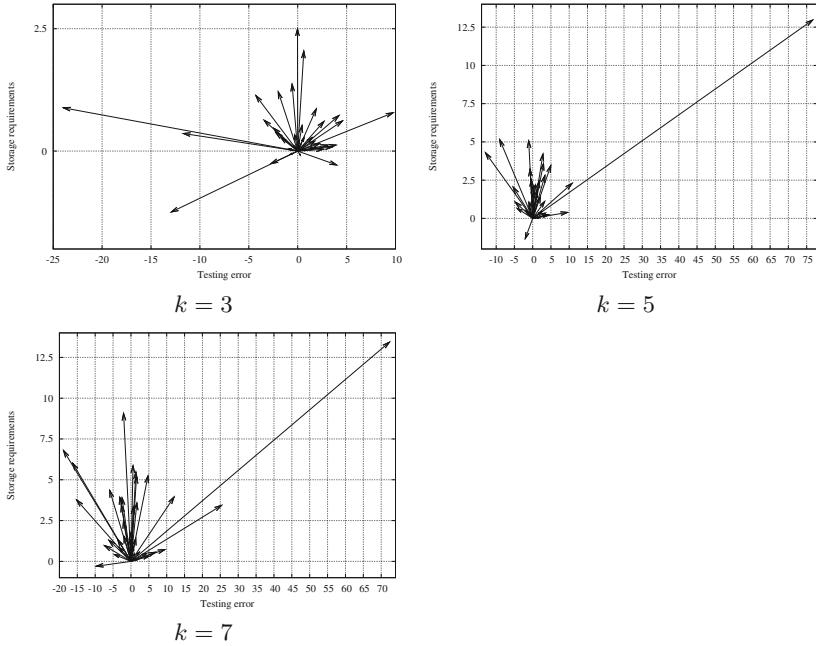


Fig. 2. Storage requirements/testing error using relative movement diagrams for class-imbalance problems

significantly better results in terms of storage and execution time and similar results in terms of the G -mean measure. We are able to match the performance of CHC, faster and with fewer instances.

5 Conclusions and Future Work

In this paper, we have presented a method for improving the performance of instance selection algorithms for class imbalanced problems, based on selecting instances more than once. For k neighbors, one instance can be selected up to $k/2$ times. With this simple modification, we achieve better storage reduction and faster execution while maintaining, and even improving, the accuracy of the instance selection process.

We must also remark that the comparison is made using as the base algorithm a CHC method, which is specifically designed for binary problems. It is likely that an evolutionary algorithm designed specifically for using integers would be even more advantageous for our proposal. However, we did not use that kind of algorithm to avoid contaminating the validation of our method.

As a future research line, we think that although this analysis tested our proposal of selecting instances more than once in evolutionary instance selection

Table 2. Comparison of CHC and our approach in terms of testing error, storage and execution time for class imbalanced methods

		Multi selection of instances		
		G -mean	Storage	Time
$k = 3$	Standard CHC	s 22/0/18	35/0/5	39/0/1
		p_s 0.6358	0.0000	0.0000
		p_w 0.5633	0.0000	0.0000
$k = 7$	Standard CHC	s 24/1/15	39/0/1	40/0/0
		p_s 0.1996	0.0000	0.0000
		p_w 0.1767	0.0000	0.0000
$k = 11$	Standard CHC	s 20/0/20	39/0/1	40/0/0
		p_s 1.0000	0.0000	0.0000
		p_w 0.7368	0.0000	0.0000

methods, this philosophy could also be applied to non-evolutionary standard instance selection methods, or to develop instance selection algorithms specifically designed for multi-selection of instances. We want also to test the scalability of our proposal [8].

References

1. Baluja, S.: Population-based incremental learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh (1994)
2. Barandela, R., Sánchez, J.L., García, V., Rangel, E.: Strategies for learning in class imbalance problems. *Pattern Recognition* 36, 849–851 (2003)
3. Basri, R., Hassner, T., Zelnik-Manor, L.: Approximate nearest subspace search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1–13 (2010)
4. Brighton, H., Mellish, C.: Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery* 6, 153–172 (2002)
5. Brodley, C.E.: Recursive automatic bias selection for classifier construction. *Machine Learning* 20(1/2), 63–94 (1995)
6. Cano, J.R., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. *IEEE Transactions on Evolutionary Computation* 7(6), 561–575 (2003)
7. 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)
8. de Haro-García, A., García Pedrajas, N.: A divide-and-conquer recursive approach for scaling up instance selection algorithms. *Data Mining and Knowledge Discovery* 18(3), 392–418 (2009)
9. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
10. Eshelman, L.J.: The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. Morgan Kaufman, San Mateo (1990)
11. Frank, A., Asuncion, A.: Uci machine learning repository (2010)

12. Franti, P., Virtajoki, O., Hautamaki, V.: Fast agglomerative clustering using a k-nearest neighbor graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(11), 1875–1881 (2006)
13. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: *Proc. of the Thirteenth International Conference on Machine Learning*, Bari, Italy, pp. 148–156 (1996)
14. Fu, Z., Robles-Kelly, A., Zhou, J.: Milis: Multiple instance learning with instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (in press, 2011)
15. García, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary Computation* 17(3), 275–306 (2009)
16. García-Osorio, C., de Haro-García, A., García-Pedrajas, N.: Democratic instance selection: a linear complexity instance selection algorithm based on classifier ensemble concepts. *Artificial Intelligence* 174, 410–441 (2010)
17. García-Pedrajas, N., Romero del Castillo, J.A., Ortiz-Boyer, D.: A cooperative co-evolutionary algorithm for instance selection for instance-based learning. *Machine Learning* 78, 381–420 (2010)
18. García, S., Fernández, A., Herrera, F.: Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. *Applied Soft Computing* 9, 1304–1314 (2009)
19. Gates, G.W.: The reduced nearest neighbor rule. *IEEE Transactions on Information Theory* 18(3), 431–433 (1972)
20. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison–Wesley, Reading (1989)
21. Kubat, M., Holte, R., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. *Machine Learning* 30, 195–215 (1998)
22. Liu, H., Motoda, H.: On issues of instance selection. *Data Mining and Knowledge Discovery* 6, 115–130 (2002)
23. Liu, J., Hu, Q., Yu, D.: A comparative study on rough set based class imbalance learning. *Knowledge-Based Systems* 21, 753–763 (2008)
24. Louis, S.J., Li, G.: Combining robot control strategies using genetic algorithms with memory. In: Angeline, P.J., McDonnell, J.R., Reynolds, R.G., Eberhart, R. (eds.) *EP 1997. LNCS*, vol. 1213, pp. 431–442. Springer, Heidelberg (1997)
25. Marchiori, E.: Class conditional nearest neighbor for large margin instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(2), 364–370 (2010)
26. Maudes-Raedo, J., Rodríguez-Díez, J.J., García-Osorio, C.: Disturbing neighbors diversity for decision forest. In: Valentini, G., Okun, O. (eds.) *Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications (SUEMA 2008)*, Patras, Grecia, pp. 67–71 (July 2008)
27. Samet, H.: K-nearest neighbor finding using maxnearestdist. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(2), 243–252 (2008)
28. Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition* 40, 3358–3378 (2007)
29. Whitley, D.: The GENITOR algorithm and selective pressure. In: *Proc 3rd International Conf. on Genetic Algorithms*, pp. 116–121. Morgan Kaufmann Publishers, Los Altos (1989)

On the Effectiveness of Distributed Learning on Different Class-Probability Distributions of Data

Diego Peteiro-Barral, Bertha Guijarro-Berdiñas, and Beatriz Pérez-Sánchez

Faculty of Informatics, University of A Coruña,
Campus de Elviña s/n, 15071 A Coruña, Spain
{dpeteiro,cibertha,bperez}@udc.es
<http://www.dc.fi.udc.es/lidia>

Abstract. The unrestrainable growth of data in many domains in which machine learning could be applied has brought a new field called large-scale learning that intends to develop efficient and scalable algorithms with regard to requirements of computation, memory, time and communications. A promising line of research for large-scale learning is distributed learning. It involves learning from data stored at different locations and, eventually, select and combine the “local” classifiers to obtain a unique global answer using one of three main approaches. This paper is concerned with a significant issue that arises when distributed data comes in from several sources, each of which has a different distribution. The class-probability distribution of data (CPDD) is defined and its impact on the performance of the three combination approaches is analyzed. Results show the necessity of taking into account the CPDD, concluding that combining only related knowledge is the most appropriate manner for learning in a distributed manner.

Keywords: Machine learning, large-scale learning, distributed learning, class-probability distribution of data.

1 Introduction

Traditionally, a bottleneck preventing the development of more intelligent systems was the limited amount of data available. However, nowadays the unrestrainable growth of data in many fields such as bioinformatics, text classification (spam, no-spam) or engineering problems such as hydroelectric or nuclear power stations, has caused that the size of the datasets is so large that the limiting factor is the inability of learning algorithms to use all the data to learn with in a reasonable time. In order to handle this problem a new field in machine learning has emerged, large-scale learning [1], where learning is limited by computational resources rather than by the availability of data. Large-scale learning intends to develop efficient and scalable algorithms with regard to requirements of computation, memory, time and communications, and has received a considerable attention in the recent years. A sample of the increasing interest generated by this topic was revealed with the organization of the workshop *PASCAL Large*

Scale Learning Challenge [2] at the *25th International Conference on Machine Learning (ICML'08)*. This workshop was concerned with the scalability and efficiency of machine learning algorithms with respect to computational time and memory resources.

Advances in network technologies have lately contributed to the proliferation of distributed systems. Consequently, in many real-world applications of machine learning, very large datasets are naturally distributed, e.g. fraud detection, market basket analysis or intrusion detection in computer networks. The best known distributed system is the Internet. The WWW stores and provides access to a prodigious amount of data. In the year 2010, its size was estimated at 5 million terabytes. Another example concerns images from earth and space telescopes. The size of such data reached the scale of exabytes and is still increasing.

As can be seen, machine learning algorithms deal more often with very large and/or distributed datasets. However, on the one hand, most current machine learning algorithms are able to deal with medium-size datasets but they cannot be applied on datasets with more than 1,000,000 data (*features* \times *samples*). Even most of them are limited to much less data due to computational time or memory restrictions. In order to overcome this issue, in practice, preprocessing techniques as subsampling are used but, and that is the point, the need for preprocessing techniques is a constraint on learning algorithms by themselves and is not a conceptual constraint for processing very large datasets. In addition to this, increasing the size of the training set of machine learning algorithms often increases the accuracy of the classification models [3]. On the other hand, most existing machine learning algorithms cannot handle distributed datasets. The most common solution comprises gathering the distributed datasets in a single location in order to merge them into a single set. However, this is often ineffective or unrealistic since the necessary central storage capacity might not be affordable (the cost of storing a single dataset in a single location is much larger than the sum of the costs of storing smaller parts of the dataset in several locations), and/or the necessary bandwidth to efficiently transmit the data to a single location might not be affordable (note also that it is common to have frequently updated databases and communication may be a continuous overhead). Even when communication cost was not too high, it is often the case that sensitive data cannot be moved around distributed locations due to privacy issues [4].

One of the most promising lines of research in order to deal with very large and/or distributed datasets is distributed learning. Distributed learning involves learning from data stored at different locations and, eventually, combining the *partial* results. On the one hand, distributed learning is able to learn from distributed datasets and, on the other hand, it is able to learn from very large datasets, since a very large dataset can be scattered across several locations. In this manner, distributed learning is inherently scalable since the growing amount of data may be offset by increasing the number of locations in which data is stored. While distributed learning seems to be the answer, only few distributed machine learning algorithms have been proposed so far in the literature [5,6].

Additionally, the assessment of most of these algorithms is performed by simulating experiments on a single computer, focusing their attention on accuracy rather than scalability. And what is more, during experimentation, most algorithms do not take into account intrinsic issues regarding distributed learning as communication costs, data privacy or *data skewness*.

This paper is concerned with a significant issue when working with distributed data, the class-probability distribution of data (CPDD). CPDD is related to the probability of occurrence of classes in a dataset and, in some manner, it is a measure of skewness of data. In a distributed system, data stored at different locations is related but the prevalence of classes may be different, e.g. the different diseases at different hospitals or buying patterns from supermarkets around the world. Up to the authors' knowledge, previous works [7] utilize other metrics for measuring difference in terms of heterogeneity of knowledge among datasets as the Euclidean distance or the Kullback-Leibler information divergence. However, they were not focused on the distributions of the classes but on the distributions of probabilities of the data. The aim here is to highlight the impact of CPDD on several approaches followed by distributed learning. The paper is structured as follows: section 2 presents the formal notion of CPDD, section 3 shows three distributed learning approaches, section 4 describes the experimental study focused on assessing the performance of such approaches, 5 shows the experimental results obtained, and sections 6 and 7 present the discussion and conclusions, respectively.

2 Class-Probability Distribution of Data

For a given subset of data, we can define CPDD as the *a priori* probability of classifying a sample in *each* class. In a less rigorous manner, CPDD is the percentage of samples per class at each location. Considering $P_i(c_k)$ and $P_j(c_k)$ as the percentage of samples of class c_k in locations i and j , respectively, and N being the number of distributed datasets or locations, we can define a *uniform* scenario as an alike CPDD for every dataset, that is,

$$\forall k, P_i(c_k) = P_j(c_k); i = 1 \dots N; j = 1 \dots N; i \neq j \quad (1)$$

Similarly, we can define a *nonuniform* scenario as an unlike CPDD for some datasets, that is,

$$\exists k, P_i(c_k) \neq P_j(c_k); i = 1 \dots N; j = 1 \dots N; i \neq j \quad (2)$$

Nonuniform scenarios should not be confused with class imbalance problems since, in this case, the unequal probability of occurrence of classes takes place intra-class, and not inter-class. Considering only two scenarios is needlessly restrictive because, firstly, it is difficult to find an absolutely uniform scenario and, secondly, this classification does not provide a degree of uniformity of a nonuniform scenario, i.e. we cannot distinguish between slight and severe nonuniform scenarios. In order to define a measure of dissimilarity between two datasets, we

introduce here the notion of distance as a measure of how different two datasets are in terms of CPDD. In this manner, C being the number of classes in the domain, the distance $d_{i,j}$ between two datasets i and j is computed as follows:

$$d_{i,j} = \sum_{k=1}^C (P_i(c_k) - P_j(c_k))^2 \tag{3}$$

If we assume $(P(c_1), P(c_2) \dots P(c_C))$ as the components of a point in the space, Eq. 3 represents the *euclidean squared distance* [8] between two points and, accordingly, it satisfies the following conditions: a) positive definiteness, b) symmetry, and c) triangle inequality [8]. This is important because any function must satisfy these condition to be considered as a function of distance.

Notice that this measure of distance is detrimental to large deviations in any class but in favor of small deviations in many classes. It is not easy to define a totally fair measure of dissimilarity between two datasets, but even so, we believe that this measure seems coherent and appropriate to quantify dissimilarities.

Based on the notion of distance between two datasets (see Eq. 3), the distance among N datasets is defined as the average distance between all pairwise datasets, that is

$$D_{AVG} = \frac{2}{N^2 - N} \sum_{i=1}^N \sum_{j=i+1}^N d_{i,j} \tag{4}$$

This measure will be useful to characterize different scenarios based on the distance among distributed datasets.

3 Learning from Distributed Data

In order to learn from distributed data, one of the most common strategies is local learning and posterior model integration [5]. This avoids moving raw data around distributed locations [9] and minimizes communication costs. In the first place, classifiers are trained on their corresponding subset of data (see Fig. 1).

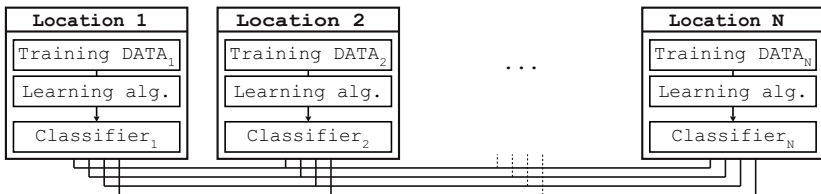


Fig. 1. Local learning at N distributed locations

Once the classifiers are trained, and analysis of the field show that three main approaches can be followed to combine them in order to obtain an unique answer for a given input (see Fig. 2)

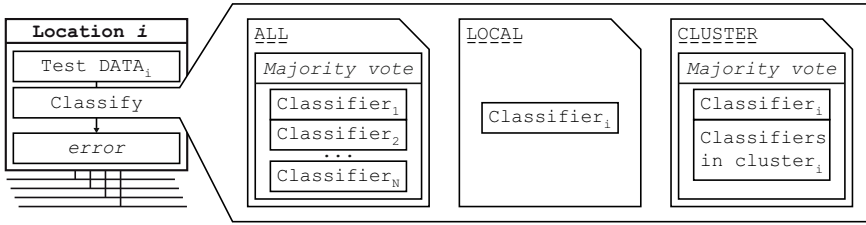


Fig. 2. Three main approaches for model integration at location i

- Use *all* classifiers to evaluate new data at all locations and then combine all answers to obtain only one. In this work, majority vote [10] was used as the combination method. It is the most popular combination method in practice and, in despite of its simplicity, it is as effective as other more complex combination methods [11] and its behavior, for the sake of this study, could be considered to be representative.
- Use *local* classifiers to evaluate new data at their respective locations. Notice that this approach does not actually learn in a distributed manner, since no knowledge is shared among classifiers. However, this approach is perfectly valid in order to learn from distributed data.

Regarding the first approach, if all datasets are considered as a single logical entity, the dissimilarities among them will not be detected. On the contrary, regarding the second approach, if all datasets are considered as different entities, the global knowledge contained in all datasets could not be detected. In order to overcome these limitations, several researchers have followed a middle way, clustering datasets and/or classifiers [12][13][14]. Using this idea, a third approach based on clustering is proposed:

- Use *similar* classifiers to evaluate test data at their respective locations, by taking a majority vote. We say that two classifiers are similar if they were trained on similar datasets (see Eq. 3), that is,

$$classifier_i \approx classifier_j \Leftrightarrow d_{i,j} \leq threshold \quad (5)$$

4 Experimental Section

The aim of this work is to assess in an experimental manner the influence of different CPDDs on the performance of the three distributed learning approaches described in Section 3. In order to do this, the materials and methods used to obtain the results are presented in this section.

4.1 Datasets

Four well-known datasets selected from the *UCI Machine Learning Repository* [15] were used. Table 1 show the characteristics of each dataset.

Table 1. Number of features (inputs), classes (outputs) and samples of each dataset

DATASET	FEATURES	CLASSES	SAMPLES
Connect-4	42	3	67,557
Coverttype	54	7	581,012
Magic	10	2	19,020
Poker hand	10	10	1,025,010

4.2 Learning Algorithms

Three of the most popular learning algorithms were used as classifiers. The algorithms were chosen in order to adequately represent different approaches in machine learning:

- *C4.5* [16] is an extension of the basic ID3 algorithm used to generate a decision tree using the concept of information entropy. The criterion for choosing a split was set to Gini’s diversity index.
- *Artificial neural networks (ANN)* [17]. Particularly, a single-layer ANN trained with the Levenberg-Marquardt backpropagation learning algorithm was used. The learning rate was set to 0.001 and it was allowed to train for a maximum of 1000 epochs.
- *Support vector machines (SVM)* [18]. Particularly, a SVM with radial basis function (RBF) kernel $\sigma = 2$ was used.

The implementation included in MATLAB[®] of these algorithms were used. Regarding the parameters’ values of the learning algorithms, it is important to remark that the question here is not which algorithm is more accurate but what is the impact of the CPDD on the performance of different combination approaches (regardless of the learning algorithms).

Finally, we are aware that there are many other techniques available in machine learning, but we believe that those selected are sufficiently representative in order to assess in an experimental manner the impact of different CPDD environments on the performance of different distributed learning approaches.

4.3 Experimental Procedure

As mentioned above, the objective of this work is the assessment of the three approaches shown in Section 3 in different environments of distributed data. With this aim, using Eq. 4 four different nonuniform CPDD scenarios were considered

$$\begin{aligned}
 \textit{None} &\rightarrow 0.00 < d_{AVG} \leq 0.10 \\
 \textit{Slight} &\rightarrow 0.10 < d_{AVG} \leq 0.15 \\
 \textit{Normal} &\rightarrow 0.15 < d_{AVG} \leq 0.25 \\
 \textit{Severe} &\rightarrow 0.25 < d_{AVG} \leq 0.50
 \end{aligned}$$

In order to truly estimate the accuracy of the three distributed learning approaches (see Section 3) for each dataset and environment, the following procedure was performed.

- Repeat $I = 100$ times
 - Divide the dataset into $N = 10$ subsets. In this manner, each subset of data represents a distributed location.
 - Divide each subset of data using *holdout validation*, i.e. a subset of samples is chosen at random to form the test set and the remaining observations are retained as the training set. The 10% of data is used for testing while the 90% are for training. This kind of validation is suitable because the size of the datasets is large.
 - Train a classifier at each location using available training data.
 - Test the classifiers at each location by calculating the standard class accuracy [19] obtained by using each of the three combination approaches described in Section 3 on unseen data.
- Compute the *mean* test accuracy and *standard deviation* of each approach over the 100 simulations.
- Apply a Kruskal-Wallis test [20] to check if there are significant differences among the medians of the approaches for a level of significance $\alpha = 0.05$.
- If there are differences among the medians, then apply a multiple comparison procedure [21] to find the approach whose error is not significantly different from the approach with the best mean accuracy rate. In this work, a Tukey’s honestly significant criterion [21] was used as multiple comparison test.

5 Experimental Results

Table 2 shows the mean test accuracies and standard deviations obtained for any of the three combining approaches. The *threshold* distance between datasets (see Eq. 3) was experimentally computed for each dataset and environment using part of the training set (validation set). Distances around 0.04 showed the best performance.

6 Discussion

As can be seen in Table 2, combining *all* classifiers reaches its best result on uniform CPDDs. This can be explained because, assuming that classifiers are interrelated, which seems coherent due to the fact that they were trained on similar data, the overall accuracy may increase as a function of the number of classifiers [22]. Regarding *local* learning, it reaches its best results on severe nonuniform CPDDs. This is a logical result due to the fact that, if data is skewed, the number of classes in some datasets may be much smaller than the number of classes in the domain, e.g. considering data from hospitals, some hospitals do not store data related to malaria because this disease has no impact on local people, but the class *malaria* exists in the domain. In this context, the smaller

Table 2. Mean test accuracies and standard deviations of each learning algorithm and combination approach on each dataset with different CPDDs, where the best results are marked in bold. Those not significantly worse than the best at the 5% confidence level are labeled with an asterisk.

DATASET	ALG.	APPR.	DEGREE OF NONUNIFORMITY			
			NONE	SLIGHT	QUITE	SEVERE
Connect-4	C4.5	All	*72.21 ± 1.75	71.47 ± 2.17	70.44 ± 2.58	68.68 ± 2.82
		Local	63.49 ± 1.96	70.62 ± 3.10	74.08 ± 3.75	76.76 ± 4.16
		Cluster	72.34 ± 1.05	74.93 ± 2.36	75.09 ± 2.64	78.72 ± 2.97
	ANN	All	74.59 ± 2.30	74.56 ± 2.89	73.08 ± 1.05	72.29 ± 1.86
		Local	*73.92 ± 1.70	75.88 ± 2.28	*80.39 ± 2.69	*81.08 ± 3.05
		Cluster	75.27 ± 1.55	78.22 ± 2.55	81.02 ± 3.07	81.63 ± 3.24
	SVM	All	76.08 ± 1.56	*75.88 ± 1.64	74.92 ± 2.72	72.86 ± 2.82
		Local	72.22 ± 2.09	73.88 ± 2.21	*78.08 ± 3.27	78.65 ± 3.20
		Cluster	74.96 ± 1.64	76.63 ± 1.84	78.22 ± 3.83	80.80 ± 3.73
Covertypes	C4.5	All	64.09 ± 2.01	63.01 ± 2.10	61.63 ± 2.67	60.87 ± 2.61
		Local	55.07 ± 2.58	57.94 ± 2.41	59.27 ± 2.57	63.81 ± 2.64
		Cluster	*63.02 ± 1.90	64.78 ± 2.80	65.41 ± 2.46	65.68 ± 2.81
	ANN	All	*56.10 ± 7.65	54.18 ± 6.08	51.63 ± 6.57	50.35 ± 8.26
		Local	39.10 ± 4.60	40.06 ± 4.58	41.41 ± 3.48	42.47 ± 2.83
		Cluster	57.20 ± 6.63	42.57 ± 2.92	41.35 ± 3.46	39.39 ± 7.07
	SVM	All	65.39 ± 1.95	65.14 ± 2.14	*64.53 ± 2.37	64.88 ± 2.38
		Local	53.33 ± 2.64	59.43 ± 1.77	60.14 ± 2.87	64.10 ± 2.62
		Cluster	*65.33 ± 2.30	*64.41 ± 2.59	66.10 ± 3.41	67.90 ± 4.25
Magic	C4.5	All	85.06 ± 1.73	83.62 ± 1.79	81.20 ± 2.04	80.77 ± 2.18
		Local	77.40 ± 2.17	80.48 ± 2.08	84.13 ± 2.63	86.19 ± 2.56
		Cluster	*84.64 ± 1.37	86.61 ± 1.97	86.85 ± 2.14	88.62 ± 2.21
	ANN	All	79.35 ± 2.36	78.94 ± 1.82	78.16 ± 2.29	77.80 ± 1.79
		Local	77.31 ± 1.44	*82.80 ± 1.89	*86.63 ± 2.22	*89.51 ± 4.38
		Cluster	77.94 ± 1.69	82.84 ± 1.82	87.06 ± 1.30	89.57 ± 4.36
	SVM	All	82.29 ± 2.53	80.61 ± 1.53	79.18 ± 2.82	79.59 ± 1.68
		Local	80.31 ± 2.26	80.63 ± 1.81	82.22 ± 4.69	*85.27 ± 2.97
		Cluster	81.47 ± 1.44	82.22 ± 0.83	83.57 ± 2.77	85.80 ± 2.41
Poker	C4.5	All	51.44 ± 1.54	50.51 ± 2.15	*50.68 ± 2.38	49.93 ± 2.63
		Local	46.44 ± 2.57	47.24 ± 2.35	*50.92 ± 2.61	52.05 ± 3.41
		Cluster	49.32 ± 1.53	*49.35 ± 2.10	51.05 ± 2.51	53.82 ± 2.97
	ANN	All	50.08 ± 1.06	49.65 ± 1.60	49.12 ± 1.59	47.49 ± 2.16
		Local	47.16 ± 1.65	*49.63 ± 2.91	52.33 ± 4.72	54.41 ± 4.10
		Cluster	49.22 ± 1.50	50.51 ± 1.76	54.76 ± 4.37	56.45 ± 4.61
	SVM	All	50.73 ± 2.62	47.37 ± 2.15	46.35 ± 2.45	44.65 ± 1.73
		Local	40.29 ± 2.03	44.86 ± 2.38	45.43 ± 3.65	46.18 ± 1.50
		Cluster	48.82 ± 1.51	49.82 ± 2.33	50.71 ± 1.97	51.33 ± 1.88

the number of classes the easier the problem of learning. Finally, *clustering* classifiers is able to exploit the advantages of both previous approaches, selecting an appropriate number of classifiers depending on the CPDD, i.e. the number of selected classifiers decreases as the nonuniformity of data increases. In this manner, clustering presents the most stable behavior with independence of the situation of non-uniformity to be handle and reaches the best performance in 34 out of 48 trials (70.83%). Moreover, it achieves a performance not significantly worse than the best approach in 5 out of 48 (10.42%). In total, clustering achieves the best, or not significantly worse than the best, performance in 39 out of 48 (81.25%). Consequently, we can assert that clustering classifiers is the most suitable approach for distributed learning regardless of the CPDD.

7 Conclusions

Distributed learning is one of the most promising lines of research for large-scale learning, since very large datasets can be scattered across several locations. In this manner, distributed learning provides a scalable solution in order to deal with very large datasets, turning an impractical algorithm into a practical one.

On the other hand, many large real-world datasets are naturally distributed. A significant issue to deal with when working with these type of distributed datasets is the CPDD. Up to the authors' knowledge, the impact of the CPDD on distributed learning algorithms has not been yet investigated in the literature, since most algorithms are simply evaluated with respect to their accuracy assuming uniform CPDD. In this manner, most existing classifier combination approaches view data distribution as a technical issue, treating distributed datasets as if they were parts of a single one. This has been identified as a very narrow view of distributed machine learning. The results obtained in this paper emphasize the necessity of taking into account the CPDD (in fact, disregarding the impact of the CPDD may lead to a significant degradation in performance). In this context, clustering classifiers before combining them is a promising line of research in order to deal with different CPDDs.

As a future work, a more sophisticated measure of distance between datasets and classifiers will be developed.

Acknowledgements. This work was supported in part by the Spanish Ministry of Science and Innovation (MICINN), Grant code TIN2009-10748, and by the Xunta de Galicia project PGIDT-08TIC012105PR, both partially supported by FEDER funds.

References

1. Bottou, L., Bousquet, O.: The tradeoffs of large scale learning. In: Advances in Neural Information Processing Systems, vol. 20, pp. 161–168 (2008)
2. PASCAL Large Scale Learning Challenge (2008), <http://largescale.first.fraunhofer.de/> (Online; accessed May 10, 2011)

3. Catlett, J.: Megainduction: machine learning on very large databases. PhD thesis, School of Computer Science, University of Technology, Sydney, Australia (1991)
4. Tsoumakas, G.: Distributed Data Mining. In: Database Technologies: Concepts, Methodologies, Tools, and Applications, pp. 157–171 (2009)
5. Tsoumakas, G., Vlahavas, I.: Effective stacking of distributed classifiers. In: Proc. 15th European Conference on Artificial Intelligence (ECAI 2002), pp. 340–344. Ios Pr. Inc. (2002)
6. Guijarro-Berdiñas, B., Martínez-Rego, D., Fernández-Lorenzo, S.: Privacy-Preserving Distributed Learning Based on Genetic Algorithms and Artificial Neural Networks. In: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, pp. 195–202 (2009)
7. McClean, S., Scotney, B., Greer, K., Páircéir, R.: Conceptual Clustering of Heterogeneous Distributed Databases. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 46–55. Springer, Heidelberg (2001)
8. Bronshtein, I.N., Semendyayev, K.A., Hirsch, K.A.: Handbook of mathematics. Springer, Berlin (2007)
9. Agrawal, R., Srikant, R.: Privacy-preserving data mining. *ACM Sigmod Record* 29(2), 439–450 (2000)
10. Dietterich, T.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
11. Lam, L., Suen, C.Y.: A theoretical analysis of the application of majority voting to pattern recognition. In: Proceedings of the 12th ICPR, vol. 2, pp. 418–420. IEEE (1994)
12. Tsoumakas, G., Angelis, L., Vlahavas, I.: Clustering classifiers for knowledge discovery from physically distributed databases. *Data & Knowledge Engineering* 49(3), 223–242 (2004)
13. Yang, W., Huang, S.: Data privacy protection in multi-party clustering. *Data & Knowledge Engineering* 67(1), 185–199 (2008)
14. Adhikari, A., Rao, P.R.: Efficient clustering of databases induced by local patterns. *Decision Support Systems* 44(4), 925–943 (2008)
15. Frank, A., Asuncion, A.: UCI machine learning repository (2010), <http://archive.ics.uci.edu/ml> (Online; accessed May 10, 2011)
16. Quinlan, J.R.: C4. 5: programs for machine learning. Morgan Kaufmann (1993)
17. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, USA (1995)
18. Vapnik, V.N.: The nature of statistical learning theory. Springer, Heidelberg (2000)
19. Weiss, S.M., Kulikowski, C.A.: Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems. Morgan Kaufmann, San Francisco (1991)
20. Hollander, M., Wolfe, D.A.: Nonparametric statistical methods (1999)
21. Hsu, J.C.: Multiple comparisons: theory and methods. Chapman & Hall/CRC (1996)
22. Hansen, L.K., Salamon, P.: Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(10), 993–1001 (1990)

On the Learning of ESN Linear Readouts

Carlos M. Alaíz and José R. Dorronsoro

¹ Departamento de Ingeniería Informática

² Instituto de Ingeniería del Conocimiento,
Universidad Autónoma de Madrid, 28049 Madrid, Spain
{carlos.alaiz,jose.dorronsoro}@uam.es

Abstract. In the Echo State Networks (ESN) and, more generally, Reservoir Computing paradigms (a recent approach to recurrent neural networks), linear readout weights, i.e., linear output weights, are the only ones actually learned under training. The standard approach for this is SVD-based pseudo-inverse linear regression. Here it will be compared with two well known on-line filters, Least Minimum Squares (LMS) and Recursive Least Squares (RLS). As we shall illustrate, while LMS performance is not satisfactory, RLS can be a good on-line alternative that may deserve further attention.

Keywords: ESN, LMS, RLS, Pseudo-inverse, Linear Regression.

1 Introduction

Multilayer Perceptrons, i.e., feed-forward neural networks (FFNN), are currently the workhorses of many classification and modelling applications, having obtained clear successes in many such problems. However, they are hard to evolve in their present form and while there is a continuous flow of research results about them, it is also clear that they suffer a kind of “paradigm fatigue”. The most obvious source for new ideas would probably be recurrent neural networks (RNN). In fact, initial work on RNN was done almost in parallel with that on FFNN, with Backpropagation Through Time (BPTT; [13]) or Real Time Recurrent Learning (RTRL; [14]) being two examples. But the results they provided were not comparable to those of FFNN, as they suffered of problems such as the existence of bifurcations where the gradient is ill-defined, exponentially gradient fading (which prevents from capturing long-term dependencies) or high computational costs, so that they can only be applied to small networks.

Partial improvements appeared around the year 2000. First, Atiya and Parlos developed a general framework to study RNN training, the so called Atiya-Parlos Recurrent Learning (APRL; [1]), that contained BPTT and RTRL as particular cases. More or less simultaneously Echo State Networks (ESN) were proposed by H. Jaeger [7] and Liquid State Machines (LSM) by J. Maass [11]. While addressing different problems (time series processing for ESN, spike train computations for LSM), both have a similar nature, in which fixed weight values and a sparse structure are proposed for hidden unit connectivity and only

linear readout weights are learned. Moreover, J. Steil showed how APRL could be connected with ESNs [12]. The fixed recurrent structure is usually called a reservoir, as it somehow stores the past behaviour of the time series under study. Because of this, the area is commonly called Reservoir Computing (RC).

However, while promising, there are still quite a few open problems. A first important issue is the definition of the reservoir architecture. Heuristic procedures are followed for ESNs, with random weight initialization and some control of the spectral radius of the connectivity matrix. For LSM, architectures are defined either based on neuroanatomic principles or seeking an edge-of-chaos behaviour [9]. These issues will not be addressed here, and the more or less standard ESN architecture definition will be followed.

A second problem in RC is the linear readout training. For ESNs it can be seen as a linear regression problem over the successive states of the reservoir and the desired outputs. The standard approach is to store these states after a wash-out period and then to derive the readout weights by the pseudo-inverse solution of the corresponding minimum squares problem. This usually results in good models provided an adequate reservoir architecture has been defined. A drawback is the batch nature of the process, that requires to store a rather long reservoir “state sample”, particularly inconvenient in time series processing. The natural alternative would be on-line methods such as Widrow’s LMS [6] gradient descent. However, LMS is not satisfactory in general, as its convergence is rather slow even when the sample covariance matrix is regular, and may even fail when it is not (singularity or near-singularity of the reservoir matrix is rather common). All this usually leads to much worse models than those obtained by pseudo-inverse computations (where covariance singularity can be controlled). An intermediate alternative is Recursive Least Squares (RLS) learning, that at each step computes the least squares solution up to that step and, theoretically, gives the pseudo-inverse solution when training ends. RLS would in principle combine the on-line nature of LMS with the good models yielded by the pseudo-inverse (but also its higher cost).

In this paper, these three algorithms for obtaining the readout weights will be compared. More precisely, in Section 2 ESN structure will be briefly reviewed and the algorithms will be described. Some experiments illustrating their behaviour will be presented in Section 3. Finally, some conclusions will be given in Section 4; basically, the results will show that LMS is not an adequate method for the on-line training of ESN readouts but, on the other hand, RLS gives adequate on-line models that are competitive with those obtained in batch mode through the pseudo-inverse. While its on-line nature is an advantage, RLS has a quadratic cost per iteration, which results in the same computational cost than the pseudo-inverse. Thus, it will be of interest to study the application to RC training of methods close to RLS but with linear cost per iteration.

2 Reservoir Computing and Readout Learning

First, the ESN-RC paradigm will be briefly reviewed. Formally, ESN networks can be divided into input, hidden and output units. Input units can be

connected to hidden and outputs, but do not receive connections from them. Hidden units (the reservoir) have recurrent connections from themselves and also from outputs. Finally output units may receive the outputs of both inputs and the reservoir. A network with L inputs, N hidden units and O outputs can be thus defined by the weights that connect the input with the reservoir $w_{\text{in}} \in \mathbb{R}^{N \times L}$, the output with the reservoir $w_{\text{f}} \in \mathbb{R}^{N \times O}$, the reservoir with the reservoir itself $w_{\text{r}} \in \mathbb{R}^{N \times N}$, and the readout weights connecting the reservoir with the output $w_{\text{o}} \in \mathbb{R}^{N \times O}$. At each time step, the RNN is updated with the equation:

$$x(i+1) = f(w_{\text{r}}x(i) + w_{\text{f}}y(i) + w_{\text{in}}u(i+1)) ; y(i+1) = w_{\text{o}}^T x(i+1) + b, \quad (1)$$

where $u(i) \in \mathbb{R}^L$ stands for the inputs at time i , $x(i) \in \mathbb{R}^N$ are the values of the N internal units at time i , $y(i) \in \mathbb{R}^O$ is the output produced by the system at time i and f is a non-linear transformation. Hidden unit bias are introduced using a constant input equal to 1. For simplicity, we will consider only one output, $O = 1$, although the results presented can be easily extended to the case of multiple outputs.

Under the RC paradigm, only the output weights w_{o} are computed. The rest of the parameters (w_{r} , w_{f} and w_{in}) are initialized randomly and kept fixed (more on this below). In the training phase, the RNN is iterated using Equation (1) and assuming that $y(i) = d(i)$, with $d(i)$ the desired output at time i (this technique is known as teacher forcing). The resulting states $x(i), i = 1, \dots, T$ are collected into a matrix $X = (x(1), \dots, x(T))^T$ whose i -th row is the state of the network at time i (assuming a perfect estimation of the output). So the problem is now to obtain each $y(i)$ as a linear combination of the components of $x(i)$. With matrix notation and denoting by $D = (d(1), \dots, d(T))^T$ the vector or matrix of desired outputs, the objective is to find a couple w_{o}, b such that $D \approx Xw_{\text{o}} + b$. This is usually formulated as trying to minimize the mean square output error (MSE).

The output weights can greatly affect the stability of the RNN (that acts as a dynamical system), producing extreme behaviours, mainly a chaotic one (in which the values produced by the network increase and oscillate uncontrollably) or a damping one (in which the networks tends to a stable constant value). This behaviour is usually related to the weight vector norm. Finally, concerning weight initialization, it should be recalled that the ESN performance is strongly determined by the sparseness of the reservoir matrix w_{r} and its spectral radius. The former determines the ESN connectivity and so its capacity for producing different (more or less independent) dynamics. The last one fixes somehow the speed of the network, in the sense that the smaller it is, the faster the network will forget the previous inputs (i.e., the ESN will only have very short term memory). When the spectral radius is close to 1 (but less to it, because greater values can cause instability), the RNN retains past information for a longer time.

As mentioned, we consider three readout weight computation algorithms. The first approach is just to apply Linear Regression using the Pseudo-inverse in a batch manner. We shall denote this pseudo-inverse least squares method as LS. The vector w_{o}^* that minimizes the MSE over the training set is given by $w_{\text{o}}^* = X^+$, where X^+ denotes the pseudo-inverse of the matrix X . Usually it is computed

through the singular value decomposition (SVD), which will numerically require a threshold ϵ to be set and discard as zero smaller eigenvalues. Obviously, this threshold may greatly affect SVD and readout computation. In practice, this method is simplest one, as it does not require any parameter (except for ϵ , which typically depends on the implementation) and usually provides stable results and good models. In contrast, it is computationally expensive, with a complexity $O(TN^2)$, i.e., linear in the number of T patterns and quadratic in the number N of units, and it only can be used in a batch context.

This complexity and lack of flexibility makes of interest on-line algorithms for linear regression. The simplest (and less expensive) one is the Least Mean Squares (LMS) filter [6] that uses an approximation of the error gradient using only instantaneous information. It is a stochastic gradient descent method, in which the weights w_o, b will be initialized to some value $w_o(0), b(0)$ and then updated for each pattern $x(i)$ depending on the error and a learning rate μ . The cost of this algorithm is linear in the dimension and linear in the number of patterns. It has been proved that its convergence depends strongly on the eigenvalue spread of the covariance matrix [4]. The learning rate μ has to be decreased if the difference between the lowest and higher eigenvalues (in norm) increases. This means that, when the covariance matrix is nearly singular, the LMS filter must perform very short steps, and thus requires a very big number of epochs. The behaviour of this algorithm when the covariance matrix is not full rank is analyzed in [3], where it is concluded that there is a component of the weight vector which remains constant during training and equal to its value at initialization. While this component does not influence training error, it may result in too large readout weights, i.e., in highly oscillating networks. When noise or rounding errors are considered, this can be a problem but in this work this case will not be considered, and w_o will be initialized to the zero vector, $w_o(0) = 0$, in order to get rid of this component. In summary, the LMS algorithm is an on-line method computationally very cheap, but it suffers from problem of convergence. The learning rate has to be adjusted depending on the eigenvalues of the covariance matrix, which are unknown in an on-line context. Even with a correct learning rate, convergence is very slow, and usually requires many epochs, making this approach almost infeasible in the ESN context, as illustrated in Section 3.

Thus, other on-line, more robust alternatives must be considered and a second on-line approach can be the Recursive Least Squares (RLS) filter [6]. RLS is not based on stochastic approximation but computes at each step an exact solution (one that minimizes the MSE) through a recursive procedure. Basically, it starts considering only one pattern, and derives the solution for the $i + 1$ -th pattern from the already computed solution using i patterns. The error function to be minimized is a weighted sum of the errors over the first i timesteps, i.e., $\varepsilon(i) = \sum_{j=1}^i \lambda^{i-j} \|e(j)\|^2$, with λ a time forgetting factor. When $\lambda = 1$, this expression coincides with standard square error, so this is the case that will be considered in this paper. Applying now the normal equations in matrix form, an optimal solution $w_o^*(i)$ that minimizes this error will satisfy $\Phi(i)w_o^*(i) = z(i)$, where

$$\Phi(i) = \sum_{j=1}^i \lambda^{i-j} x(j)x(j)^T, \quad z(i) = \sum_{j=1}^i \lambda^{i-j} x(j)d(j).$$

After some derivations, it can be proved [6] that $w_o^*(i) = w_o^*(i-1) + K(i)\xi(i)$, where $P(i) = \Phi(i)^{-1}$, $\xi(i) = d(i) - x(i)^T w_o^*(i-1)$ and

$$K(i) = \frac{\lambda^{-1}P(i-1)x(i)}{1 + \lambda^{-1}x(i)^T P(i-1)x(i)}.$$

Thus, the algorithm will update at each step $K(i)$ and $\xi(i)$, and then use them to compute $w_o^*(i)$ and $P(i)$. The complexity of RLS will then be the same than that of LS, due to the quadratic complexity of each step. As it will be shown with the experiments, it gives rather RNN good models.

We sum up the previous model discussions. The LS algorithm provides stable and good unique solutions for regular matrices and even singular ones. On the other hand, it is costly and may produce solutions with a high norm in singular problems, due to the mentioned requirement of a threshold ϵ to discard near zero eigenvalues. Moreover, it can not be used in an on-line manner. The LMS algorithm is on-line and it has cheap steps but it depends on the initial conditions, although this can be controlled. However, it also requires to carefully choose a learning rate that depends on the eigenvalues of the covariance matrix, and usually has convergence problems for singular or near singular covariance matrices. Finally, the RLS algorithm gives ideally the same solution as LS, but in an on-line context. Although the numerical approximations prevent it from getting the exact pseudo-inverse solution, it usually yields good models. Nevertheless, it is as complex as LS.

3 Numerical Experiments

In this section the preceding algorithms will be compared over four problems. The first and simplest is the construction of a sinewave generator. The problem has no explicit inputs, although teacher forcing is used during training and the previous target value enters the network dynamics through the feedback connections. The desired output at time i is simply given by $d(i) = \frac{1}{2} \sin\left(\frac{i}{4}\right)$. The training set is formed by 700 timesteps, that is, about 11 cycles. The next 300 steps will be used for testing. As the results will show, this turns out to be a rather easy problem. Our second example is a well known problem for ESNs, a modulated sinewave, which consists on the generation of a sinewave whose frequency changes with the inputs. Formally, the single input of the problem at time i is $u(i) = \frac{\sin(0.01\pi i)+1}{2}$, and the corresponding output is $d(i) = \frac{\sin(a(i))+1}{2}$, where $a(i) = a(i-1) + 0.1 + 0.9u(i)$, and $a(0) = 0$. In this case, 2000 timesteps will be used for training, and the following 1000 for testing. In the third example the semi-chaotic trajectory

of Mackey–Glass attractor will be predicted. This temporal series is the solution [5] of the differential equation

$$d'(t) = -0.1d(t) + \frac{0.2d(t-17)}{1+d(t-17)^{10}}.$$

Here the system inputs during training are the previous 10 delays. Testing will be done in a generative manner, using the model predictions as input in subsequent steps. The step length is $\Delta t = 1$, the training set consists of 3000 timesteps (about 60 “cycles”, where by a cycle we mean a piece of the trajectory between two consecutive maxima); the testing set consists of the following 3000 steps. Finally, our last example is based on the 3–dimensional trajectory of the Roessler Attractor [10], given by the coordinates of the vector $(z_1(t), z_2(t), z_3(t))$ satisfying the differential equations

$$z'_1 = -z_2 - z_3, \quad z'_2 = z_1 + 0.2z_2, \quad z'_3 = 0.2 + z_1z_3 - 5.7z_3.$$

The coordinates z_1 and z_3 will be used as inputs, i.e., we take $u(i) = (z_1(i), z_3(i))$; z_2 will be the desired output, $d(i) = z_2(i)$. The step length is $\Delta t = 0.1$. A set of 1000 steps (about 17 cycles) will be used to train the model, and the next 1000 to test it.

To compare the algorithms, two measures will be considered. The first one is the Normalized Root MSE (NRMSE) over the testing set, i.e., $\text{NRMSE} = 100 \frac{\text{MSE}}{\sigma_d}$, where σ_d is the standard deviation of the desired output. In other words, NRMSE represents the error as a percentage of the signal’s standard deviation. While this is a basic quality measure, it has the drawback that the ESN can diverge when running for a long time, distorting NRMSE values. For this reason, two NRMSE values will be computed considering only the first one and three test cycles to obtain an idea of the evolution of the trained RNN. Nevertheless, the NRMSE can still be misleading and we introduce as a second measure the number of steps before the trajectory output diverges from the desired output more than a threshold θ . We call this measure Acceptable Evolution Length (AEL), which we define as $\text{AEL}_\theta = \min_i \|d(i) - y(i)\| > \theta$. This can be understood as the length that the model keeps a “good” trajectory. In our experiments we will take $\theta = \frac{1}{2}\sigma_d$.

Since RNNs under the RC paradigm are strongly dependent on the initialization, we will consider 50 randomly initialized different ESNs. Since this leads to outliers, the mean and standard deviation of the previous measures can result in misleading values, so instead the median and median absolute deviation (the median of the absolute distance of each result to the median) will be included as robust variants of the mean and standard deviation respectively.

Regarding the configuration of the RNN, the sparseness of the reservoir has been controlled using only non zero weight values on 10%, 20% or 30% of the reservoir connections. For each problem we consider the sparseness value giving best results, namely 30% for Sinewave and Roessler, 20% for Mackey–Glass and 10% for Modulated Sinewave. Two different reservoir sizes N will be explored for each problem, a first one with acceptable results and twice this. After scaling, the spectral radius will be 0.6 in Roessler and 0.8 for the other problems.

Table 1. Measures for the simulations (RMSE as percentage of the standard deviation and AEL_θ with $\theta = \sigma_d$)

	LS	LMS	RLS	LS	LMS	RLS
	$N = 50$			$N = 100$		
Sinewave for $S = 30\%$						
NRMSE ₁	0.005 ± 0.003	60.68 ± 49.33	<i>0.087</i> ± 0.072	0.002 ± 0.001	33.37 ± 20.04	<i>0.032</i> ± 0.022
NRMSE ₃	0.013 ± 0.011	91.37 ± 29.35	<i>0.141</i> ± 0.125	0.008 ± 0.005	62.25 ± 43.78	<i>0.066</i> ± 0.060
AEL _θ	300.0 ± 0.0	23.0 ± 20.8	300.0 ± 0.0	300.0 ± 0.0	48.5 ± 46.7	300.0 ± 0.0
	$N = 150$			$N = 300$		
Modulated Sinewave for $S = 10\%$						
NRMSE ₁	0.062 ± 0.036	24.52 ± 11.40	<i>0.211</i> ± 0.105	0.005 ± 0.003	23.41 ± 3.849	<i>0.076</i> ± 0.028
NRMSE ₃	0.062 ± 0.040	38.19 ± 15.91	<i>0.208</i> ± 0.121	0.006 ± 0.003	37.83 ± 5.292	<i>0.081</i> ± 0.035
AEL _θ	168.0 ± 63.8	12.0 ± 5.9	168.0 ± 54.9	344.0 ± 269.8	16.0 ± 5.9	<i>189.5</i> ± 58.6
Mackey–Glass for $S = 20\%$						
NRMSE ₁	1.667 ± 1.045	58.95 ± 45.61	<i>3.337</i> ± 2.237	0.244 ± 0.163	46.85 ± 29.44	<i>1.746</i> ± 0.851
NRMSE ₃	<i>61.88</i> ± 86.11	146.9 ± 80.61	18.29 ± 14.57	2.023 ± 1.952	120.3 ± 59.34	<i>7.646</i> ± 4.035
AEL _θ	<i>109.5</i> ± 58.6	16.0 ± 12.6	139.5 ± 59.3	204.5 ± 88.2	18.0 ± 4.4	202.5 ± 39.3
Roessler for $S = 30\%$						
NRMSE ₁	76.26 ± 98.06	76.23 ± 49.36	62.40 ± 89.47	210.7 ± 110.1	48.44 ± 13.15	<i>107.1</i> ± 107.6
NRMSE ₃	164.9 ± 116.0	167.6 ± 32.52	165.6 ± 122.4	284.7 ± 100.6	160.4 ± 27.84	<i>223.1</i> ± 54.20
AEL _θ	33.5 ± 24.5	28.0 ± 13.3	34.5 ± 24.5	21.5 ± 8.9	29.5 ± 5.9	26.0 ± 10.4

The training process is as follows. First, the reservoir is randomly initialized and then iterated over the training inputs using teacher forcing to obtain the reservoir matrix X . The first 10% rows of X are discarded to “forget” their dependence on the initial state of the RNN. The vector D is given as part of the training set. Then, 3 outputs weights, one for each of the algorithms above, will be computed independently. This allows to obtain paired results, making easier posterior comparisons.

Our LS algorithm uses the ϵ value standard in the *Octave* package implementation of the pseudo-inverse (that depends on the greatest singular value). For RLS, $\lambda = 1$ will be used to have the same cost function of LS and LMS. For LMS, the learning rate will be specified as $\mu = \frac{0.2}{\text{trace}R}$ that theoretically guarantees convergence, and the samples will be fed through 100 epochs. For testing, each of the three RNN (differing in the output weights) will be iterated over a test set, using when needed previous step outputs as next step inputs. The NRMSE and AEL values will be registered and a Wilcoxon rank test will be applied at a 10% significance level to check whether these distributions are different. When they are so, an algorithm will be considered as better if its error median is smaller or its AEL median larger. The best results will be show in **bold** face, and the second better in *italic*. When no statistical difference exists between two or the three distribution, they will be shown in the same font face.

Table 1 shows NRMSE and AEL values. For the Pure Sinewave the LS algorithm has the best NRMSE error, but recalling that errors are given as percentages of the standard deviation, both LS and RLS capture very well the sinewave evolution. This is confirmed by the AEL measure, which indicates that both models complete 300 steps in almost all of the 50 trials. By contrast, LMS performs very poorly, achieving big errors and very short acceptable trajectories. In the three algorithms, the results improve with the increment of the reservoir size. The histogram of the AEL distributions for this problem (not shown) confirmed

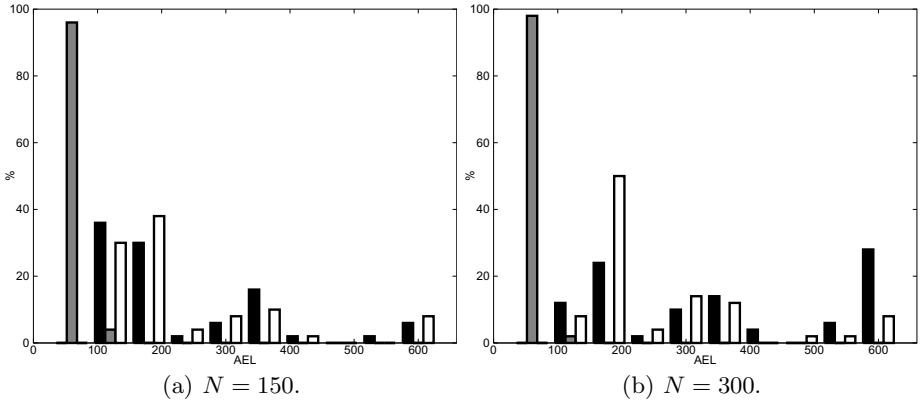


Fig. 1. Histogram for the modulated sinewave problem, comparing LS (*black*), LMS (*grey*) and RLS (*white*) AEL distributions. The connectivity is $S = 10\%$.

this analysis, showing that both LS and RLS perform very well. In contrast, the LMS distribution shows a worse behaviour, that only slightly improves for the bigger reservoir.

The situation for the Modulated Sinewave problem is similar to the sinewave case. Table 1 shows that the best NRMSE values are those of LS, although both LS and RLS have very low errors. LMS has again a worse performance. Looking at AEL behaviour, for $N = 150$ there is a draw between LS and RLS. For $N = 300$, both results improve but LS is now clearly better. LMS again performs much worse. The AEL histograms are presented in Figure 1, where it can be seen that LS and RLS produce some very good models with an acceptable length of about 600 steps (a better model has larger right histogram values). When the reservoir is bigger, LS beats RLS.

For Mackey–Glass Table 1 shows that for $N = 150$, LS presents a better error in the first cycle, but is beaten by RLS over three cycles, which has a good evolution (in fact, LS error is five times bigger in the third cycle). For $N = 300$ LS wins on NRMSE, followed by RLS. Again, LMS performance is much worse. RLS has a better AEL for the smaller reservoir followed closely by LS. For the larger reservoir, both obtain the same results. In Figure 2 the histograms of this problem are presented. When increasing the number of units, the distributions of LS and RLS move to the right, both producing very good models with more than 400 steps of acceptable trajectory, although LS produces more models of this type.

The last rows of Table 1 contain the Roessler errors. With a small reservoir ($N = 150$), the three algorithms obtains similar errors. Although LMS performs is slightly worse, this difference is not significant. With $N = 300$ LMS beats the other two algorithms; in this case, both LS and RLS worsen over the bigger RNN, while LMS improves its errors. In terms of AEL, the better models for the small network are LS and RLS, while for the big reservoir LMS and RLS become the best models; nevertheless, the results are in general poor. Although

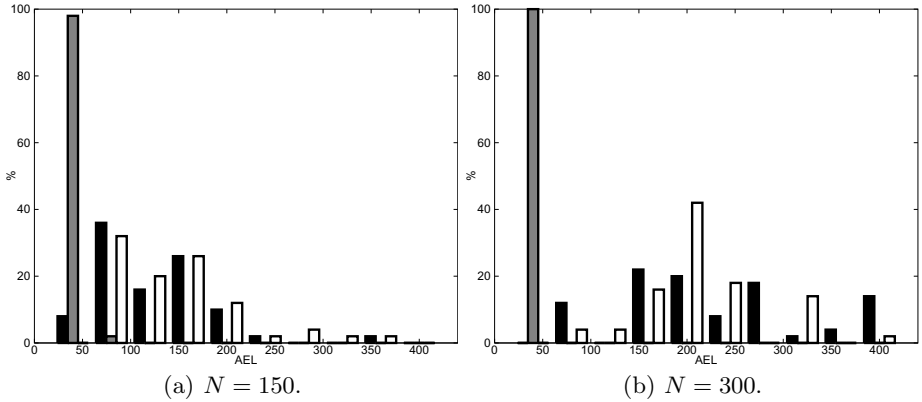


Fig. 2. Histogram for the Mackey–Glass problem, comparing LS (*black*), LMS (*grey*) and RLS (*white*) AEL distributions. The connectivity is $S = 20\%$.

not shown, the histograms for this problem show that LS and RLS produce a few very good models which keep a good trajectory over more than 400 steps. Curiously, and probably caused by some over-fitting, the number of good models decreases when increasing the reservoir size. LMS has a very stable and narrow distribution.

Summing things up, RLS performance is equal or better than that of LS in terms of the AEL, although it tends to achieve less precise models when measured by RMSE. On the other side, LMS produces poor results due in general to convergence problems.

4 Conclusions

In this paper, three different and well known linear regression algorithms are analyzed and compared for Reservoir Computing readout training. For this task, the well known algorithm is standard regression using the pseudo-inverse (LS). To overcome its high complexity and batch nature, the LMS filter would be a natural alternative as an on-line method with low complexity. Nevertheless, LMS has well known convergence problems and, as shown experimentally, in practice it does not obtain acceptable solutions even when a huge number of epochs is used. As an intermediate option, the RLS filter has been considered. Although being an on-line method, it yields models comparable to the LS method, specially in terms of the length of the acceptable prediction trajectories. Its drawback is its large complexity, equal to that of LS. Nonetheless, the results here indicate a good potential for the RLS approach and suggest the consideration of other on-line filters with a linear cost per step. One option could be the so called transversal filters [2], but they are usually designed for problems with a special “near-to-Toeplitz” correlation matrix. Another option are the gain adaptation

algorithms (compared in [§](#) with RLS and others). These and other related issues are currently under consideration.

Acknowledgement. The authors acknowledge partial support from grant TIN2010-21575-C02-01 of the TIN Subprogram of Spain's MICINN and of the Cátedra UAM-IIC en Modelado y Predicción. The first author is also supported by the FPU-MEC grant AP2008-00167 and kindly thanks Graz University of Technology for receiving him during a visit supported by FPU-MEC grant reference AP2008-00167ESTANCIA-2010.

References

1. Atiya, A., Parlos, A.: New Results on Recurrent Network Training: Unifying the Algorithms and Accelerating Convergence. *IEEE Transactions on Neural Networks* 11, 697–709 (2000)
2. Cioffi, J., Kailath, T.: Fast, recursive-least-squares transversal filters for adaptive filtering. *IEEE Transactions on Acoustics, Speech and Signal Processing* 32(2), 304–337 (1984)
3. Eweda, E.: Convergence analysis of adaptive filtering algorithms with singular data covariance matrix. *IEEE Transactions on Signal Processing* 49(2), 334–343 (2001)
4. Feuer, A., Weinstein, E.: Convergence analysis of LMS filters with uncorrelated Gaussian data. *IEEE Transactions on Acoustics, Speech and Signal Processing* 33(1), 222–230 (1985)
5. Glass, L., Mackey, M.: Mackey-Glass equation. *Scholarpedia* 5(3), 6908 (2010)
6. Haykin, S.: *Adaptive Filter Theory*. Prentice Hall, New Jersey (2001)
7. Jaeger, H.: Echo state network. *Scholarpedia* 2(9), 2330 (2007)
8. Lanzi, P., Loiacono, D., Wilson, S., Goldberg, D.: Prediction update algorithms for XCSF. In: *Proceedings of GECCO 2006*, pp. 1505–1512 (2006)
9. Legenstein, R., Maass, W.: Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks* 20(3), 323–334 (2007)
10. Letellier, C., Rossler, O.: Rossler attractor. *Scholarpedia* 1(10), 1721 (2006)
11. Maass, W., Natschläger, T., Markram, H.: Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation* 14(11), 2531–2560 (2002)
12. Steil, J.: Backpropagation–Decorrelation: online recurrent learning with $O(N)$ complexity. *IEEE Transactions on Neural Networks* 2, 843–848 (2004)
13. Werbos, P.: Backpropagation Through Time: What It Does and How to Do It. *Proceedings of the IEEE* 78(10), 1550–1560 (1990)
14. Williams, R., Zipser, D.: A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation* 1, 270–280 (1989)

Learning Naive Bayes Models for Multiple-Instance Learning with Label Proportions

Jerónimo Hernández and Iñaki Inza

Intelligent Systems Group, University of the Basque Country
{jeronimo.hernandez, inaki.inza}@ehu.es

Abstract. This paper deals with the problem of multi-instance learning when label proportions are provided. In this classification problem, the instances of the dataset are divided into disjoint groups, where there is no certainty about the labels associated with individual samples. However, in each group the number of instances that belong to each class is known. We propose several versions of an EM-algorithm that learns naive Bayes models to deal with the exposed problem. The proposed algorithms are evaluated on synthetic and real datasets, and compared with state-of-the-art approaches. The obtained results show a competitive behaviour of our proposals.

Keywords: supervised classification, Multiple-instance learning with label proportions, EM algorithm, Naive Bayes.

1 Introduction

The term Multiple-Instance Learning (MIL) refers to a supervised classification problem where the instances are grouped, the individual instance labels are unknown and there is a unique label per group [1]. In MIL, this group label is positive if at least one instance of the group is positive, and negative otherwise. In this paper, we deal with a variation of this problem where a count of the classes of the instances is given for each group, i.e. it is known how many instances of each class are in each group. The objective remains to classify any new example.

There are many real problems that fit into this framework. This situation is usually found in cases in which the relation between instance and label is lost for some reason; this may be due to privacy preserving or non-monitoring process. One such case is that of election votes, where some parties stand for institutions and, in each zone, each party gets an exact number of votes. The election results are known, but it is unknown for which party each citizen voted. Other real problems include embryo selection in assisted reproductive technology [8], spam filtering, e-commerce, etc.

The problem of *Multiple-Instance Learning with Label Proportions* (MIL-LP) can be formally defined as follows: the dataset (D) that describes this problem is composed of m examples $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$. These examples are provided

grouped in n bags—or sets of examples— \mathbf{B}_i ($D = \{\mathbf{B}_1 \cup \mathbf{B}_2 \cup \dots \cup \mathbf{B}_n\}$ such that $\mathbf{B}_i \cap \mathbf{B}_j = \emptyset, \forall i \neq j$). For each bag \mathbf{B}_i , the number of instances, m_i , is known (such that $m = \sum_{i=1}^n m_i$), as well as the *counts* m_{ic} , i.e. the number of instances in \mathbf{B}_i that have the label c such that $\sum_{c \in C} m_{ic} = m_i$ where C is the set of class labels. Note that bag label information can also be provided in terms of proportions [4], $p_{ic} \in [0, 1]$ where $\sum_{c \in C} p_{ic} = 1$.

In this problem, types of bags can be distinguished: bags with certainty in the label of the instances, and bags without certainty. On the one hand, if all the instances in bag \mathbf{B}_i belong to the same class, then individual labels of the instances in \mathbf{B}_i are known. In this kind of bag, known as *full bag*, there exists a class label c such that $m_{ic} = m_i$. On the other hand, bags usually have instances that belong to different classes, which implies that there is uncertainty in the label of the instances. In this kind of bags, which are known as *non-full bags*, for all possible class c , $m_{ic} < m_i$.

There exist in the literature several methods to deal with the MIL-LP problem. The first time that a method was proposed to learn from this kind of data was in [2], where Kück et al. present a MCMC strategy. Later, other methods have been proposed, such as MeanMap [3][4], basic adaptations of KNN, ANN, SVM and DT [5], Kernel K-means [6] and another version of SVM [7].

The rest of the paper is organised as follows. In the next section, three new versions of an Expectation-Maximization (EM) method for learning a naive Bayes model for MIL-LP are proposed. Then, the experiments are presented: an analysis on artificial data that evaluates the efficacy of our proposals in different experimental conditions, and a comparison with state-of-the-art approaches. The paper finishes with some conclusions and future work.

2 Learning a Naive Bayes Model for MIL with Label Proportions: An Expectation-Maximization Method

In this paper, we develop several EM strategies to learn naive Bayes models for MIL-LP. A naive Bayes model is a probabilistic classifier that assumes conditional independence between the predictive variables given the class variable. This assumption allows the classifier to be defined using the maximum a posteriori (MAP) estimate as:

$$\hat{c} = \underset{c}{\operatorname{argmax}} p(C = c) \prod_{i=1}^v p(X_i = x_i | C = c)$$

Note that in this model, the only parameters that should be learnt are the class priors $p(c)$ and the conditional probabilities $p(x_i | c)$.

Learning a naive Bayes implies the estimation of the model parameters from the available dataset. In a classical dataset where all the instances have a known individual class label, all model parameters can be estimated with maximum likelihood estimates by means of frequency counts. When part of the class labels are missing, as in MIL-LP, some techniques can be used to learn the model parameters despite the uncertainty.

Consequently, we use an Expectation-Maximization (EM) [9] method. This is an iterative procedure that can be used to obtain the maximum likelihood parameters in the presence of missing data. It can also be used to obtain the MAP estimate or to fill missing data. Each iteration consists of two steps, expectation (E) and maximization (M). The E-step estimates the missing data as the conditional expectation of the likelihood given the current fit for the model parameters. In the M-step, the model parameters are re-estimated such that the likelihood is maximized given the data completed in the E-step. Convergence is guaranteed since EM increases the likelihood at each iteration.

2.1 An EM Method for MIL with Label Proportions

We use an EM algorithm to learn naive Bayes models for MIL-LP problems. Our EM method estimates the individual labels of the instances in non-full bags (missing data) at the E-step and, at the M-step, learns a naive Bayes model for the previously completed dataset.

However, in the MIL-LP problem, there is some information about the class labels of non-full bags, the label proportions. Therefore, we can not use a standard EM algorithm as there are forbidden label assignments (those that do not fulfill the label proportions). Moreover, it is possible to take advantage of the label proportions in order to learn more accurate models. In order to complete the data, the EM method should take into account all the possible assignments (*valid complexions*), that are exactly as many as a multinomial coefficient:

$$\binom{m_i}{m_{i1} \dots m_{ic}} = \frac{m_i!}{\prod_{c \in C} m_{ic}!}$$

The individual assignment of labels is not independent, i.e. the assignment of a label to an instance in a non-full bag affects other assignments in the bag. The probability of a valid complexion of the labels in a bag needs to be calculated as the joint probability of the labels assigned to the instances in the bag. In order to calculate the probability of a label of an instance, the probability of the complexions that assign that label to the instance are added up.

We propose three different EM versions, which share the M-step and only differ in the way in which they complete the data:

PEM_{MIL-LP}. For each instance in a non-full bag, this approach calculates the probability of each possible label, as has been explained before. We call this approach probabilistic.

NPEM_{MIL-LP}. In this approach, we do not consider a probabilistic complexion of the labels of non-full bags. Instead, for each instance, the label of the complexion with the highest probability is assigned. We call this approach non-probabilistic.

MCEM_{MIL-LP}. This approach also carries out a probabilistic complexion, but uses a Markov Chain Monte Carlo (MCMC) procedure to obtain it [9]. This is an iterative procedure that approximates the expectation of a hidden variable. It performs this by sampling iteratively the variable, where each sample

is a modification of the previous one. This sequence of samples is expected to reach a steady state, where the mean of the samples converges to the expected value of the hidden variable. It requires two numeric parameters, the *burn-in* and the number of samples. The first indicates the number of samples removed in the beginning of the sampling in order to make the approximation more reliable (it is supposed to not have reached the steady state yet). The other parameter indicates the number of samples that are generated to approximate the expectation. This approach implements a rejection MCMC procedure. Rejection means that, during the sampling process, a new sample can be rejected if its probability is lower than the probability of the previous sample. If it is rejected, the previous complexion is repeated to replace it. Next, the sampling continues.

In our method, the samples are valid complexions of the label proportions of a non-full bag. At each step, a new sample is obtained by swapping the positions of two randomly chosen (and different) labels of the current complexion. The probability of a class label c for an instance in a bag is calculated as the proportion of samples that assign that label to the instance. This approach has been developed because covering all the valid complexions, as the two previous approaches do, becomes unfeasible when bag size grows.

3 Experiments

The experiments have two objectives: evaluating our proposals when dealing with different experimental conditions and comparing them with other state-of-the-art approaches.

In order to carry out a comparison with previous methods, the conditions of the experiments have been reproduced and the same real datasets have been selected from two public repositories, UCI [10] and LibSVM [11].

Regarding the evaluation of our proposals, artificial data has been generated to test the behaviour of our three versions of EM when dealing with different sizes of bag (m_i) and distributions of labels in the bags. A dataset of 100 instances has been sampled from a naive Bayes model (with 5 binary predictive variables and one binary class variable). The parameters of the model are generated randomly sampling a Dirichlet distribution with all the parameters equal to 1.

The EM method has only one parameter, the stop condition, which has been fixed to 200 iterations. $\text{MCEM}_{\text{MIL-LP}}$ also requires another two parameters that are fixed to 100 iterations of burn-in and another 1000 iterations to approximate the label expectation. Continuous variables of real datasets are discretized in 3 equal-frequency intervals.

3.1 Validation in MIL with Label Proportions

In a real problem of MIL-LP, instances of the dataset are provided grouped in bags and, some of these (non-full bags), are indivisible for the validation process

because of the uncertainty in the labels (i.e. different labels in a bag makes it impossible to know, a priori, which specific label each instance has).

Therefore, the adaptation of classical validation techniques (cross-validation, training/test, bootstrap...) to this problem is not straightforward. In particular, cross-validation (CV) requires performing a division of the dataset in folds such that the number of instances in all the folds is the same. Making such a division in folds by respecting the bags and, at the same time, trying to keep the same number of instances in each fold is not straightforward. In fact, this problem can be seen as a generalization to more than two subsets of the classical combinatorial optimization problem called “number partitioning”.

This issue is even more complicated if divisions are also required to be stratified, as a new condition is introduced in the optimization: the combination of the label proportions of the bags in a fold has to fulfill the global label proportions of the dataset.

In this paper, a simple solution to solve this optimization problem is used. First, non-full bags are ordered according to their size, from bigger to smaller. Then, at each step, the instances of the corresponding non-full bag are assigned to the most empty fold. In the case of a tie, the fold is chosen randomly. Once all the non-full bags have been distributed, the instances in full bags (which are divisible because of their label homogeneity) are used to compensate the number of instances in the folds and the stratification requirement.

Due to the lack of public availability of real MIL-LP datasets, it is common to use artificial datasets or real datasets that are not MIL-LP [4] [5], and transform them into a problem of this type by building (or aggregating) bags in order to validate the algorithms. *Aggregation* is the process in which the instances of a dataset are somehow grouped in bags and, for each bag, the label proportions of its instances are calculated (bag label) and the instances are separated from their labels.

In order to validate a model for a non-real MIL-LP dataset, there exist two strategies for integrating the aggregation and validation processes.

The first strategy, which is the only one used in the related literature, is to divide the dataset for validation in a classical way, and then, to aggregate bags in the training data. This is a simplification that only takes into account the procedure of learning from label proportions, putting aside other related problems. It is not a realistic approach because the aggregation depends on a previous validation division. For example, in a CV process, bags are aggregated at each CV iteration, so the validation process is performed with different bags at each step. Moreover in this strategy, if the validation is class-stratified, the individual label information (unknown in non-full bags) is used to perform the division.

The second strategy consists of aggregating bags in the dataset from the beginning. Once bags have been aggregated, the dataset can be considered as a real MIL-LP dataset. Now, it has to be learned using techniques of real MIL-LP problems, including the optimization problem in validation presented above.

3.2 Evaluation of Proposed Methods with Artificial Data

In this section, the three versions of our EM method are evaluated in terms of accuracy and computational time, dealing with different sizes of bag (m_i) and label distributions in the bags.

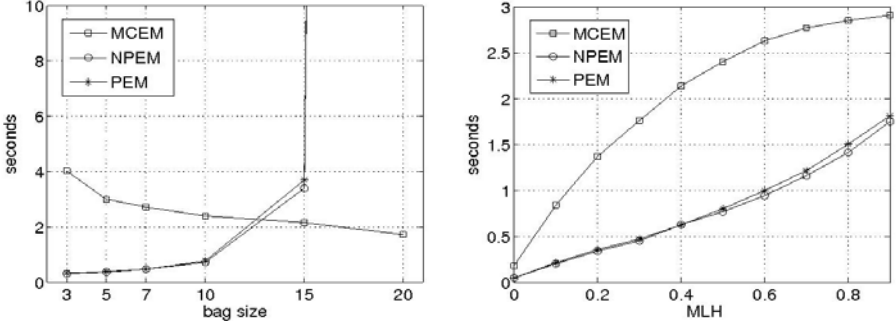


Fig. 1. Left figure: computational time needed by the three versions of the proposed algorithm to accomplish a 5-fold CV with $MLH = 0.5$ and increasing bag size (m_i). Right figure: the same process is performed with $m_i = 10$ and increasing MLH .

In order to measure the uncertainty of the distribution of the labels in bags of a dataset, we calculate the *mean label entropy*, MLH . This measure represents the mean, for all the bags, of the entropy of the label proportions in each bag.

Since tested domains are non-real MIL-LP datasets, we can aggregate bags such that the MLH value of the dataset reaches some desired value. The minimum MLH is obtained when the dataset is ordered by the class of the instances and bags are aggregated with contiguous instances. Based on this, by swapping two instances that have different labels and are in different bags, the MLH value can be modified. Then, a simple way to configure a dataset with a specific level of entropy is to swap instances until this level is reached. MLH is mapped into the interval $[0, 1]$ in order to do it more comprehensible. The maximum MLH value is reached when all bags fulfill the global label proportions of the dataset.

We have tested our proposals in terms of the computational time.

In order to observe the influence of m_i , the three proposals are evaluated for $m_i = \{3, 5, 7, 10, 15, 20\}$ and the results are presented in Fig. 1a, which shows the mean of the computational time spent by a 5-fold CV process over 20 repetitions. After CV splitting, bags are aggregated with MLH equals to 0.5.

In order to observe the influence of MLH , the three versions of our method are evaluated for $MLH = \{0.0, \dots, 0.9\}$, with a step of 0.1. Fig. 1b shows the mean of the computational time spent by a 5-fold CV process over 20 repetitions. After CV splitting, bags are aggregated with $m_i = 10$.

As expected, bag size makes the versions PEM_{MIL-LP} and $NPEM_{MIL-LP}$ unfeasible for $m_i > 15$ (the number of complexions is exponential to the bag size). Instead, $MCEM_{MIL-LP}$ is faster as m_i grows because, for a constant size

of the dataset, larger m_i implies a lower number of bags. Remember that our $\text{MCEM}_{\text{MIL-LP}}$ version executes the same MCMC procedure for each non-full bag.

The higher the MLH, the more computationally expensive our methods are (higher entropy implies more non-full bags). However, in the case of $\text{MCEM}_{\text{MIL-LP}}$, once all bags have become non-full, time will not increase even if MLH rises. With the other two proposed versions the behaviour is different; more entropy implies more complexions and, since these versions explore all the valid complexions, it always demands more computational time.

Once we know the limits of our method, its three versions are evaluated when dealing with different experimental conditions, increasing the bag size for $m_i = \{2, 3, 5, 7, 10, 12, 15\}$ and the entropy for $\text{MLH} = \{0.0, \dots, 0.9\}$, with a step of 0.1. All this procedure has been performed twice for both the exposed strategies of integrating aggregation and validation. Figure 2 shows the results, which represent the mean accuracy of a 5-fold CV process over 20 repetitions.

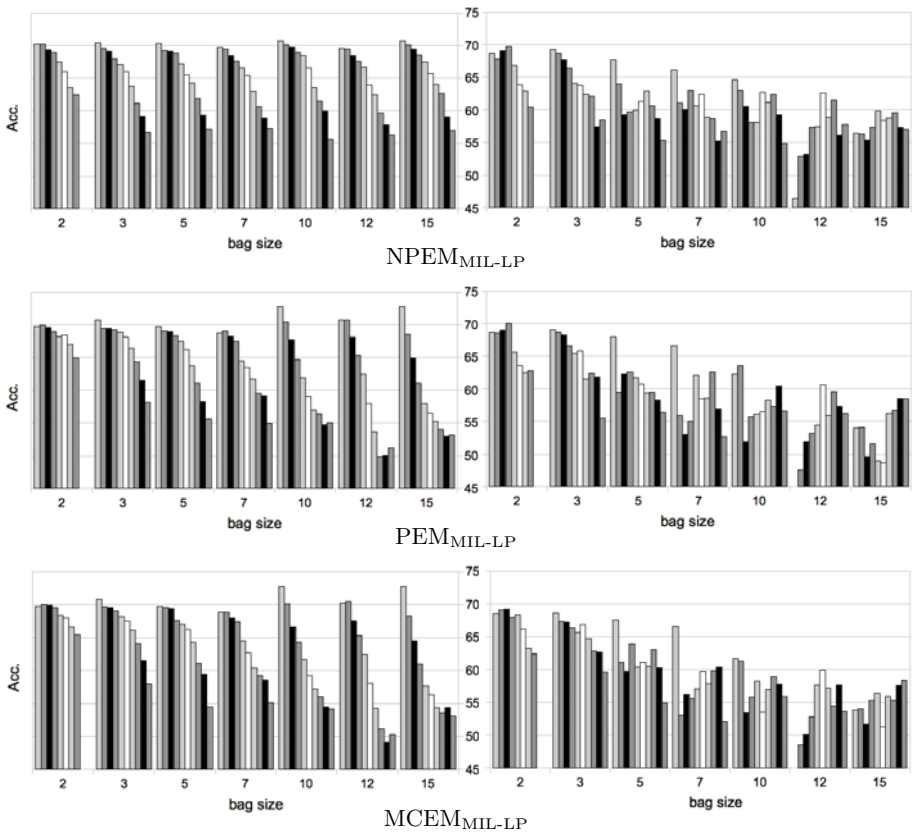


Fig. 2. Mean accuracy by a 20x5CV process for different bag sizes (m_i). For each m_i , the bars inside show the variation of the results for increasing MLH. Left figure: experiments that follow the first strategy of integrating aggregation and validation. Right figure: experiments that follow the second strategy.

Due to the fact that reaching $MLH = \{0.8, 0.9\}$ with $m_i = 2$ has been unfeasible, the results for these conditions are omitted.

In the figures, the results of the second strategy of integrating aggregation and validation are shown to be worse and not as uniform as the results of the first strategy. These results demonstrate that MIL-LP is a problem that, apart from the learning process, involves many challenges. It seems that the second strategy deals with some problem that the first strategy skips; it is probably the division for accuracy estimation of a dataset with instances aggregated in bags.

In the first strategy, the $NPEM_{MIL-LP}$ seems to be less affected by the increasing of MLH and, even more important, its results show a low variation for different bag sizes (m_i). The behaviour of the other two versions, PEM_{MIL-LP} and $MCEM_{MIL-LP}$, are very similar. In both cases, the larger the bags, the higher the difference in accuracy between low and high values of MLH. In the second strategy, the results of the $NPEM_{MIL-LP}$ are again those that are less affected by the modification of the bag size (m_i) and the MLH.

3.3 Comparison with State-of-the-Art Approaches

In order to check the robustness of our proposals, we have reproduced the experiments of Musicant et al. [5] and Quadrianto et al. [4] with real datasets. The comparison is performed in terms of accuracy; this is because it is the only available results of their methods.

The first comparison is performed with respect to the proposals of Musicant et al. [5]. The authors evaluate their techniques (basic adaptations to MIL-LP of KNN, ANN, SVM and Decision Trees) under different experimental conditions using 5-fold CV. The first strategy of integrating aggregation and validation is performed (first CV, and then, bags are aggregated).

In order to generate different experimental conditions, two parameters are modified: bag size and randomness. The last parameter represents a concept similar to our MLH term. The randomness parameter indicates the number of swaps over the label-ordered dataset that have to be performed to modify the distribution of labels in the bags. The number of swaps takes the values $r = \{0, 25, 50, 100, 200, 500, 2000\}$ and the bag size $m_i = \{2, 5, 10, 20\}$.

Due to lack of space, only one UCI dataset [10] is used, Breast-Cancer Wisconsin. The results are shown in Table 1. Musicant et al. [5] indicate that their methods could be tuned in order to improve the results. The same holds for our methods. However, the objective of these experiments is to show how different conditions of the MIL-LP problem affect the learning process of each method. This can only be achieved by fixing the base classifier parameters and modifying only the MIL-LP problem parameters (bag size and randomness). In this way, we can see that our proposals and those of Musicant et al. show a similar behavior.

Note that the results of PEM_{MIL-LP} and $NPEM_{MIL-LP}$ for $m_i = 20$ are omitted because it is unfeasible to run these versions with m_i larger than 15.

The second comparison is performed with respect to the proposals of Quadrianto et al. [4]. The authors carry out different experiments in two scenarios: the number of bags being equal to the number of classes ($n = |C|$), and the

Table 1. Breast-Cancer W. dataset. Accuracy for increasing the bag size (m_i) and number of swaps (r). Left table: proposals of Musicant et al. Right table: our proposals.

	$m_i \backslash r$	0	25	50	100	200	500	2000
K-NN	2	0,97	0,97	0,96	0,96	0,94	0,92	0,90
	5	0,97	0,97	0,96	0,96	0,94	0,92	0,90
	10	0,97	0,97	0,96	0,96	0,94	0,92	0,90
	20	0,97	0,96	0,95	0,93	0,84	0,66	0,66
ANN	2	0,91	0,91	0,90	0,89	0,87	0,84	0,84
	5	0,91	0,89	0,85	0,83	0,70	0,65	0,66
	10	0,89	0,87	0,84	0,78	0,68	0,66	0,65
	20	0,89	0,88	0,85	0,79	0,68	0,64	0,64
SVM	2	0,94	0,95	0,95	0,95	0,96	0,95	0,95
	5	0,96	0,95	0,95	0,95	0,96	0,95	0,93
	10	0,95	0,94	0,94	0,96	0,95	0,96	0,93
	20	0,94	0,96	0,91	0,94	0,92	0,95	0,93
DT	2	0,95	0,95	0,95	0,95	0,95	0,95	0,95
	5	0,95	0,95	0,95	0,95	0,95	0,95	0,95
	10	0,95	0,95	0,95	0,95	0,94	0,95	0,95
	20	0,95	0,95	0,96	0,95	0,94	0,95	0,94

	$m_i \backslash r$	0	25	50	100	200	500	2000
NPEM	2	0,97	0,97	0,97	0,97	0,97	0,97	0,97
	5	0,97	0,97	0,97	0,97	0,97	0,97	0,97
	10	0,97	0,97	0,97	0,97	0,97	0,96	0,95
	20	-	-	-	-	-	-	-
PEM	2	0,97	0,97	0,97	0,97	0,97	0,97	0,97
	5	0,97	0,97	0,97	0,97	0,97	0,97	0,97
	10	0,97	0,97	0,97	0,97	0,97	0,97	0,97
	20	-	-	-	-	-	-	-
MCEM	2	0,97	0,97	0,97	0,97	0,97	0,97	0,97
	5	0,97	0,97	0,97	0,97	0,97	0,97	0,97
	10	0,97	0,97	0,97	0,97	0,97	0,97	0,97
	20	0,97	0,97	0,97	0,97	0,97	0,97	0,97

number of bags being larger than the number of classes. In both cases, 10-fold CV is performed as a part of the first strategy of integrating aggregation and validation.

In the first scenario, two bags are aggregated by dividing the dataset in two class-stratified groups. After this, all the instances of the second class are removed from the first group and then, bags are aggregated. The result is a dataset with a full bag (all the instances of the same class) and a non-full bag with the same label proportions as the original dataset. The authors present the results of their method, MeanMap, and also the results of the proposal of Kück et al. [2].

Table 2. Mean classification error and associated standard deviation. In the left part, first scenario. Results of MeanMap (Quadrianto et al. [4]), MCMC (Kück et al. [2]), and MCEM_{MIL-LP} for six UCI/LibSVM datasets. In the right part, second scenario. Results of unweighted and weighted MeanMap (Quadrianto et al. [4]) and MCEM_{MIL-LP} for three LibSVM datasets.

Dataset	MeanMap	MCMC	MCEM	unweighted	weighted	MCEM
australian	17, 0 ± 1, 7	30, 8 ± 1, 8	23, 76 ± 0, 52	34, 44 ± 4, 03	29, 58 ± 3, 71	24, 97 ± 0, 23
breast c.w.	5, 3 ± 0, 8	4, 8 ± 2, 0	3, 75 ± 0, 17	----	----	----
heart	30, 0 ± 4, 0	33, 7 ± 4, 7	21, 57 ± 0, 66	----	----	----
ionosphere	18, 4 ± 3, 2	18, 0 ± 2, 1	19, 3 ± 0, 92	----	----	----
splice	25, 2 ± 2, 0	28, 8 ± 1, 6	24, 0 ± 0, 91	33, 43 ± 1, 65	21, 12 ± 2, 59	21, 67 ± 0, 39
svmguide3	20, 4 ± 0, 9	24, 2 ± 0, 8	41, 88 ± 0, 82	24, 28 ± 2, 20	18, 5 ± 1, 73	37, 31 ± 0, 26

In the second scenario, which Quadrianto et al. call “overdetermined system”, 8 bags are aggregated following the label proportions that are exposed in [4]. The authors present the results of two versions of their proposal, called weighted and unweighted MeanMap.

We showed in the previous section that the only version that can deal with large bag size is $\text{MCEM}_{\text{MIL-LP}}$. Since the bags in these experiments are huge, only $\text{MCEM}_{\text{MIL-LP}}$ has been used. The results, which are shown in Table 2, vary depending on the datasets. In spite of this difference, our methods seem to be competitive with respect to the proposals of Kück et al. and Quadrianto et al.

4 Conclusions

In this paper we have proposed three competitive versions of an EM method to learn naive Bayes models for MIL problems when label proportions are provided. Starting from a basic adaptation of the EM methodology to learn simple Bayesian network classifiers, the work has been demonstrated to be a competitive starting point for further developments.

For future work, the performance with more complex Bayesian network models or with non-binary class datasets can be tested. Also, model parameters for individual datasets can be tuned. It would be interesting to integrate our three EM versions into a unique EM method. In order to estimate the labels of a specific bag in the E-step, the method could choose the most suitable strategy attending to the characteristic of the bag and take advantage of the benefits of using each proposal. For example, while $\text{MCEM}_{\text{MIL-LP}}$ is the best option for bags with high MLH, for bags with low MLH, it could be better to use $\text{NPEM}_{\text{MIL-LP}}$ because of its tolerance to different conditions of the problem.

References

1. Dietterich, T., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence* 89(1-2), 31–71 (1997)
2. Kück, H., de Freitas, N.: Learning about individuals from group statistics. In: *Proc. 21th Conference on Uncertainty in Artificial Intelligence*, pp. 332–339 (2005)
3. Quadrianto, N., Smola, A.J., Caetano, T.S., Le, Q.V.: Estimating labels from label proportions. In: *Proc. 25th International Conference on Machine Learning*, New York, pp. 776–783 (2008)
4. Quadrianto, N., Smola, A.J., Caetano, T.S., Le, Q.V.: Estimating labels from label proportions. *Journal of Machine Learning Research* 10, 2349–2374 (2009)
5. Musicant, D.R., Christensen, J.M., Olson, J.F.: Supervised learning by training on aggregate outputs. In: *Seventh IEEE International Conference on Data Mining*, pp. 252–261 (2007)
6. Chen, S., Liu, B., Qian, M., Zhang, C.: Kernel K-means based framework for aggregate outputs classification. In: *2009 IEEE International Conference on Data Mining Workshops*, pp. 356–361 (2009)
7. Rueping, S.: SVM Classifier Estimation from Group Probabilities. In: *Proc. 27th International Conference on Machine Learning* (2010)

8. Morales, D.: Clasificadores Bayesianos en la Selección Embrionaria en Tratamientos de Reproducción Asistida. PhD thesis, University of the Basque Country (2008)
9. McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions. Wiley Series in Probability and Statistics. Wiley-Interscience (2008)
10. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, Irvine, <http://archive.ics.uci.edu/ml>
11. Fan, R.: LIBSVM Data: Classification, Regression and Multi-label. National Taiwan University, <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

The von Mises Naive Bayes Classifier for Angular Data

Pedro L. López-Cruz, Concha Bielza, and Pedro Larrañaga

Computational Intelligence Group
Departamento de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo
Boadilla del Monte, 28660, Madrid, Spain
pedro.lcruz@upm.es, {mcbielza,pedro.larranaga}@fi.upm.es

Abstract. Directional and angular information are to be found in almost every field of science. Directional statistics provides the theoretical background and the techniques for processing such data, which cannot be properly managed by classical statistics. The von Mises distribution is the best known angular distribution. We extend the naive Bayes classifier to the case where directional predictive variables are modeled using von Mises distributions. We find the decision surfaces induced by the classifiers and illustrate their behavior with artificial examples. Two applications to real data are included to show the potential uses of these models. Comparisons with classical techniques yield promising results.

Keywords: Naive Bayes classifier, supervised classification, circular statistics, directional statistics, angular data, von Mises distribution.

1 Introduction

Scientists from a wide range of fields use angles to capture some properties of the phenomena they study, e.g., meteorologists analyze the direction of wind currents and waves, biologists measure the growth direction of plants and the movement of animals, etc.

Angular data have some distinctive properties that rule out the use of classical statistics. Therefore, common descriptive statistical tools have to be adapted to work with this kind of information, e.g., rose diagrams are used instead of regular histograms, the mean direction is computed taking into account the periodicity of the data, etc. Directional statistics [1,2] provides the theoretical background and the techniques to properly manage these data.

In this paper, we introduce the von Mises naive Bayes (vMNB) classifier for use with angular data. We review the naive Bayes classifier (NB) in Sect. 2, and the von Mises distribution in Sect. 3. Section 4 introduces vMNB, and its decision surfaces and properties are analyzed at length. Artificial examples are used to illustrate the behavior of the classifiers. Two applications to real data

and the statistical comparisons with classical techniques are included in Sect. 5. Finally, Sect. 6 concludes with a discussion and outlines future work. Detailed derivations of the formulas can be found in the Appendix.

2 The Naive Bayes Classifier

The NB classifier [3] is one of the best known models for supervised classification [4]. In NB, the class is modeled as a discrete variable C , and the set of its possible class values is noted $val(C)$. The set of predictive variables is $\{X_1, \dots, X_n\}$. Figure 1 shows the graphical structure of the NB classifier, where the nodes represent the variables in the domain, and the arcs encode the conditional (in)dependence relationships between them [5]. NB assumes that the predictive variables are conditionally independent given the class value. NB uses a maximum a posteriori decision rule to classify the objects, i.e., it assigns each object to the class c^* with maximum posterior probability. Given an object with predictive variable values $\mathbf{x} = (x_1, \dots, x_n)$, this is obtained as:

$$c^* = \arg \max_{c \in val(C)} p(C = c) \prod_{i=1}^n \rho(X_i = x_i | C = c),$$

where $\rho(\cdot)$ is a general probability function, i.e., a probability distribution $p(\cdot)$ for discrete variables or a probability density function $f_X(\cdot)$ for continuous variables.

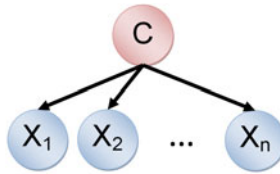


Fig. 1. Graphical structure of the naive Bayes classifier

Although conditional independence is a strong assumption, NB has been successfully applied to a wide range of problems [6], and its theoretical properties have been studied at length [7]. NB is a linear classifier when binary [3] or multinomial [8] predictive variables are used. On the other hand, the decision surfaces are polynomials when ordinal predictive variables are used [4].

3 The von Mises Distribution

The periodicity of angular data rules out the use of classical probability distributions. The most straightforward solution is to wrap linear distributions around the circle. Several distributions have been adapted according to this approach,

e.g., the wrapped normal distribution [9]. However, specific probability distributions have also been proposed for angular data. The von Mises distribution [10] is the best known circular distribution, as it is the circular analogue of the normal distribution. A variable Φ , defined in a circular domain $(-\pi, \pi]$, follows a von Mises distribution $vM(\mu_\Phi, \kappa_\Phi)$ if its probability density function is

$$f_\Phi(\phi; \mu_\Phi, \kappa_\Phi) = \frac{\exp(\kappa_\Phi \cos(\phi - \mu_\Phi))}{2\pi I_0(\kappa_\Phi)}, \tag{1}$$

where μ_Φ is the mean direction, $\kappa_\Phi \geq 0$ is the concentration of the points around the mean, and $I_\nu(\cdot)$ is the modified Bessel function of the first kind with order $\nu \in \mathbb{R}$, defined by

$$I_\nu(x) = \frac{1}{2\pi} \int_0^{2\pi} \cos(\nu\phi) \exp(x \cos \phi) d\phi .$$

The von Mises distribution is unimodal, with the mode (highest density) at μ_Φ and the antimode (lowest density) at $\mu_\Phi \pm \pi$. The distribution of the points around the circumference is uniform when $\kappa_\Phi = 0$, whereas high values of κ_Φ yield points tightly clustered around the mean. Given a sample of N points $\{\phi_1, \dots, \phi_N\}$, the maximum likelihood estimators of the parameters in the distribution are the sample mean direction

$$\hat{\mu}_\Phi = \arctan \frac{\bar{C}}{\bar{S}}, \text{ with } \bar{C} = \frac{1}{N} \sum_{i=1}^N \cos \phi_i, \text{ and } \bar{S} = \frac{1}{N} \sum_{i=1}^N \sin \phi_i, \tag{2}$$

and the sample concentration value

$$\hat{\kappa}_\Phi = A^{-1}(\bar{R}), \text{ where } A(\hat{\kappa}_\Phi) = \frac{I_1(\hat{\kappa}_\Phi)}{I_0(\hat{\kappa}_\Phi)} = \bar{R} = \sqrt{\bar{C}^2 + \bar{S}^2} . \tag{3}$$

Unfortunately, $\hat{\kappa}_\Phi$ cannot be found analytically and approximations have to be computed numerically [2]. Figure 2 shows a sample of 100 points drawn from the distribution $\Phi \sim vM(\pi/4, 5)$ using the CircStat toolbox for MATLAB [11].

4 The von Mises Naive Bayes Classifier

In this section we introduce the vMNB classifier, where the conditional probability density functions of the predictive variables are modeled using von Mises distributions. The conditional probability densities for a variable Φ given the class value c are noted $(\Phi|C = c) \equiv \Phi^{(c)} \sim vM(\mu_\Phi^{(c)}, \kappa_\Phi^{(c)})$. We study the behavior of the classifier by deriving the decision surfaces it induces. We assume that the class is binary, e.g., $val(C) = \{1, 2\}$. When the class has more than two values, we have to compute the decision surface for each pair of values and label each subregion with the class having the maximum posterior probability. For detailed derivations of the decision surfaces included in this paper see the Appendix available at http://cig.fi.upm.es/components/com_phocadownload/container/vmnbappendix.pdf.

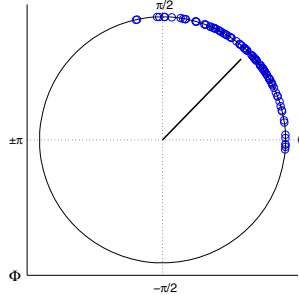


Fig. 2. Sample of 100 points drawn from the distribution $vM(\pi/4, 5)$. The black line represents the sample mean direction $\hat{\mu}_\Phi$ and its length is the sample mean resultant length \bar{R} .

4.1 One Predictive Variable

We first analyze the simplest scenario where only one predictive variable Φ is used for classification. The decision surface induced by the classifier is computed by equating the posterior probability distribution of the two class values

$$p(C = 1|\Phi = \phi) = p(C = 2|\Phi = \phi) . \tag{4}$$

By applying Bayes' rule and substituting the von Mises density (II) in (4), we get

$$\frac{p(C = 1)}{2\pi I_0(\kappa_\Phi^{(1)})} \exp(\kappa_\Phi^{(1)} \cos(\phi - \mu_\Phi^{(1)})) = \frac{p(C = 2)}{2\pi I_0(\kappa_\Phi^{(2)})} \exp(\kappa_\Phi^{(2)} \cos(\phi - \mu_\Phi^{(2)})) .$$

Simplifying, taking logarithms and operating, we finally get the two angles that bound the class subregions (see the Appendix):

$$\begin{aligned} \phi' &= \alpha + \arccos(D/T) \\ \phi'' &= \alpha - \arccos(D/T), \end{aligned}$$

where $\cos \alpha = a/T$, $\sin \alpha = b/T$, $D = -\ln \frac{p(C=1)I_0(\kappa_\Phi^{(2)})}{p(C=2)I_0(\kappa_\Phi^{(1)})}$, $T = \sqrt{a^2 + b^2}$, $a = \kappa_\Phi^{(1)} \cos \mu_\Phi^{(1)} - \kappa_\Phi^{(2)} \cos \mu_\Phi^{(2)}$, and $b = \kappa_\Phi^{(1)} \sin \mu_\Phi^{(1)} - \kappa_\Phi^{(2)} \sin \mu_\Phi^{(2)}$.

vMNB finds two angles that divide the circumference into two subregions, one for each class value. The two angles ϕ' and ϕ'' are defined with their bisector angle α , which depends on the mean directions $\mu_\Phi^{(1)}, \mu_\Phi^{(2)}$ and concentrations $\kappa_\Phi^{(1)}, \kappa_\Phi^{(2)}$, of Φ given each of the two class values. The distance between the angles also depends on both the concentration and the mean directions. Alternatively, we can substitute $(x, y) = (\cos \phi, \sin \phi)$ to compute the Cartesian coordinates of the decision surface that bounds the class subregions, obtaining the following expression (see the Appendix for details):

$$(\kappa_{\Phi}^{(1)} \mu_X^{(1)} - \kappa_{\Phi}^{(2)} \mu_X^{(2)})x - (\kappa_{\Phi}^{(1)} \mu_Y^{(1)} - \kappa_{\Phi}^{(2)} \mu_Y^{(2)})y - D = 0 . \quad (5)$$

Equation (5) defines a decision line that bounds the class regions. Therefore, vMNB with one predictive variable is a linear classifier.

We illustrate the behavior of the classifier with an artificial example. The class variable C is binary and its values are considered equiprobable a priori, i.e., $p(C = 1) = p(C = 2) = 0.5$. The conditional probability densities of Φ given each class value are $\Phi^{(1)} \sim vM(\pi/2, 2)$ and $\Phi^{(2)} \sim vM(\pi, 5)$. Figure 3(a) shows a sample of 100 points drawn from these distributions, whereas Fig. 3(b) shows the classification provided by vMNB and the decision angles that bound the class regions (green lines): $\phi' = 2.43$ (139.23°) and $\phi'' = -1.67$ (-95.63°).

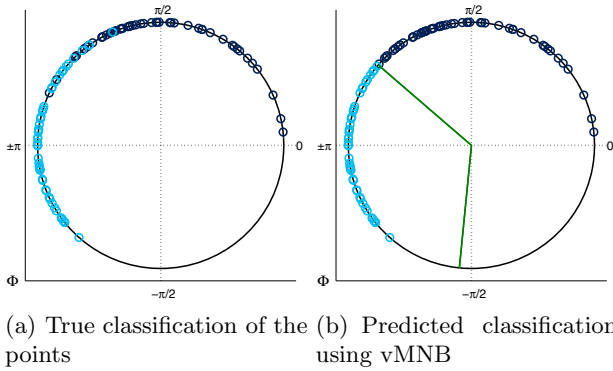


Fig. 3. True class and class predicted using vMNB for a sample of 100 points. Points with $C = 1$ are shown in dark blue, whereas points with $C = 2$ are shaded light blue.

4.2 Two Predictive Variables

We can use the same approach to analyze the behavior of the classifier when two circular predictive variables Φ and Ψ are included in the model. In this scenario, the domain defined by the predictive variables is a torus $(-\pi, \pi] \times (-\pi, \pi]$. The decision surface induced by the vMNB classifier is given by

$$p(C = 1|\Phi = \phi, \Psi = \psi) = p(C = 2|\Phi = \phi, \Psi = \psi) . \quad (6)$$

By applying conditional independence, Bayes' rule, substituting the von Mises density function (1) in (6) and operating, we get

$$a \cos \phi + b \sin \phi + c \cos \psi + d \sin \psi + D = 0, \quad (7)$$

where a, b, c, d and D are constants (see the Appendix). The Cartesian coordinates of the points lying on the surface of a torus can be computed using:

$$\begin{aligned}
 x &= (L + l \cos \phi) \cos \psi \\
 y &= (L + l \cos \phi) \sin \psi \\
 z &= l \sin \phi,
 \end{aligned}
 \tag{8}$$

where L is the distance from the center of the torus to the center of the revolving circumference that generates it, and l is the radius of the revolving circumference. Isolating the trigonometric functions in (8), replacing them in (7) and operating, we get the following decision surfaces:

$$\begin{aligned}
 clx + dly - az^2 + bz\sqrt{l^2 - z^2} + bLz + (aL + Dl)\sqrt{l^2 - z^2} + al^2 + Dll &= 0, \\
 clx + dly - az^2 - bz\sqrt{l^2 - z^2} + bLz - (aL + Dl)\sqrt{l^2 - z^2} + al^2 + Dll &= 0.
 \end{aligned}$$

These decision surfaces are quadratic in z , so vMNB is a more complex and flexible classifier when two variables are included than when only one variable is used. This behavior is different in the NB with discrete predictive variables, where the complexity of the decision surfaces (hyperplanes) remains the same when the number of predictive variables is increased [8]. The decision surfaces are also hyperplanes when the predictive variables are statistically independent and modeled with Gaussian distributions that share the same variance. However, as far as we know, no result has been given in this particular scenario, where the predictive variables are conditionally independent given the class value and have different variances.

The following artificial example illustrates this behavior. Figure 4(a) shows a sample of 1000 points drawn using the distributions $\Phi^{(1)} \sim vM(\pi, 2)$ and $\Psi^{(1)} \sim vM(-2\pi/3, 6)$ for points in class $C = 1$, and distributions $\Phi^{(2)} \sim vM(\pi/2, 5)$ and $\Psi^{(2)} \sim vM(\pi, 3)$ for points in class $C = 2$. The classes are considered equiprobable a priori. The classification provided by vMNB and the complex decision boundaries that separate the two class regions are shown in Fig. 4(b).

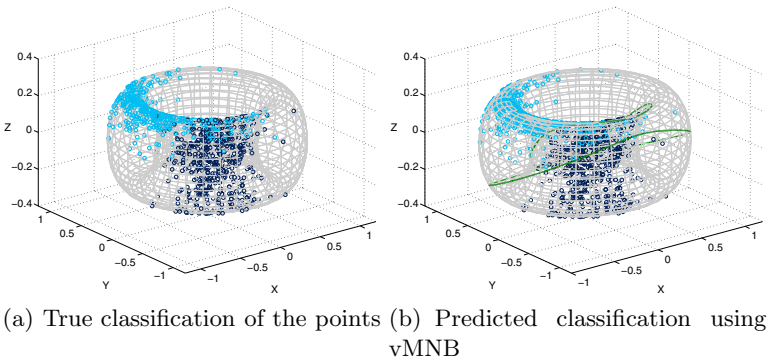


Fig. 4. True class and class predicted using vMNB for a sample of 1000 points. Points with $C = 1$ are shown in dark blue, whereas points with $C = 2$ are shaded light blue. The decision boundaries are drawn in green.

5 Experiments with Real Data

In this section, we apply the classifiers presented above to real world data from two different problems studied in biology:

Group Identification in Megaspores: In this problem we classify two groups of lycopsid megaspores based on the angle of orientation of the sporopollenin wall elements. The dataset is an example included in Oriana software (<http://www.kovcomp.co.uk/oriana>). It was first obtained and analyzed in [12]. The data are measured in degrees and represent the orientation of the element relative to a baseline drawn perpendicular to the spore surface. The two groups of megaspores used in this study are called *Selaginellalean* and *Isoetalean*. The dataset includes 960 entries, where 360 are *Selaginellalean* (37.5%) and 600 are *Isoetalean* (62.5%).

Protein Secondary Structure Prediction Using Dihedral Angles: The three dimensional structure of proteins is the key to identifying their function and behavior [13]. Many models tend to predict the protein secondary structure before modeling the tertiary structure. Dihedral angles (ϕ, ψ) are of key importance since they primarily define the protein's backbone conformation. In this example, we use the dihedral angle values of aminoacids to distinguish between the two most common secondary structures in proteins: the α -helix and the β -sheet. The data were retrieved from the Protein Geometry Database [13]. The dihedral angles for all the compositions corresponding to one residue were retrieved. We erased the instances with missing dihedral angles and selected the conformations corresponding to α -helices and β -sheets to obtain a dataset containing 49,676 instances. The number of instances for each class value were 28,141 α -helices (56.65%) and 21,535 β -sheets (43.35%). Figure 5(a) shows an α -helix (light blue) and a β -sheet (dark blue) conformation in a protein. Figure 5(b) shows the dihedral angles of all the aminoacids in α -helix (light blue) and β -sheet (dark blue) data conformations, mapped into a torus. Von Mises distributions have been used to model dihedral angles of protein structures in a number of works, e.g., [14,15].

We use vMNB to solve these problems. The maximum likelihood estimators of the parameters in Eq. (2) and (3) are computed using [11]. As far as we know, supervised classification problems using angular data as predictive information have not been systematically studied before. Therefore, we could not find any other approaches that manage directional data. We compare our results with the commonly used Gaussian NB classifier (GNB) and the multinomial NB classifier (mNB) using a supervised discretization algorithm [16]. The accuracy of the classifiers is estimated with a stratified 10-fold cross-validation procedure. Table 1 shows the classifiers' accuracies. We test if the difference in accuracy is significant by applying a right-tailed t -test over the sorted difference of accuracies in a 10-fold cross validation averaged over 10 runs, as recommended in [17]. Table 1 also shows the p-values of this t -test for each pair of classifiers (the first classifier is better than the second). In Megaspores dataset, we can only find statistical differences between GNB and mNB. On the other hand, vMNB

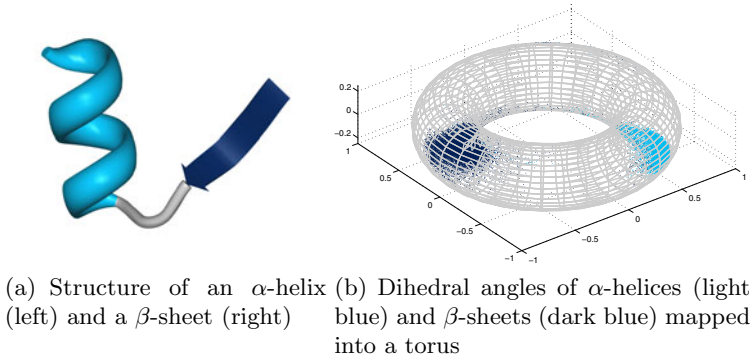


Fig. 5. Structure and dihedral angle distribution of α -helices and β -sheets structures

outperforms both GNB and mNB in Protein dataset. Figure 6 illustrates the difference between modeling protein dihedral angle ψ with a Gaussian or a von Mises conditional distribution for $C=2$. The Gaussian distribution ignores the periodicity of the data and yields different densities for angles 180° (0.24) and -180° (0.0), which refer to the same angle. Also, the von Mises distribution is more peaked. The log-likelihood for the von Mises distribution given the data is higher than for the Gaussian distribution (see the legend in Fig. 6).

Table 1. Mean accuracy and standard deviation of the classifiers computed with stratified 10-fold cross-validation (left). P-values of a right-tailed t -test to check whether the difference in accuracy is significant (right).

	vMNB	GNB	mNB	vMNB vs. GNB	vMNB vs. mNB	GNB vs. mNB
Megaspores	76.56 ± 4.26	76.46 ± 4.26	74.79 ± 5	0.7287	0.0607	0.0461
Protein	98.04 ± 0.18	97.64 ± 0.19	97.76 ± 0.24	0.0000	0.0001	0.9962

6 Discussion

In this paper, we introduced the vMNB classifier for use with angular and directional data. First, the NB classifier and the univariate von Mises distribution were reviewed. Then, we analyzed the behavior of vMNB when von Mises distributions are used to model the conditional probability distributions of the predictive variables. We derived the decision surfaces for one and two predictive variables and illustrated them with artificial examples. We showed that vMNB is a linear classifier when only one predictive variable is included. Also, we showed that the decision surfaces induced by vMNB are much more complex when two

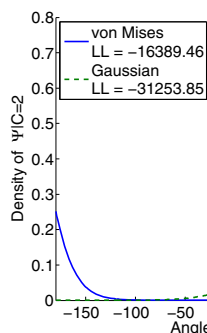


Fig. 6. Gaussian (dashed) and von Mises (solid) conditional density functions for ψ dihedral angles of class C=2 in Protein dataset

predictive variables are considered. Adding more predictive variables to vMNB can be easily done, and we could expect the complexity of the decision surfaces induced by these classifiers to grow accordingly. Two applications to real data from the field of biology were reported. The vMNB classifier achieved similar or better results than GNB and mNB in those datasets.

Conditional independence is a strong assumption, so a number of Bayesian classifiers that relax the NB assumption have been proposed, e.g., [18,19,20]. Extending vMNB to these Bayesian classifiers is not a straightforward matter. On the one hand, the conditional mutual information between variables modeled with von Mises distributions has to be computed in [19,20]. On the other hand, both marginal and conditional distributions of a multivariate von Mises cannot be von Mises distributions [21], making it difficult to model statistical dependencies between angular variables. Estimating the parameters of multivariate von Mises distributions is also challenging.

Hybrid scenarios combining discrete and continuous variables occur frequently in science. Classification models including categorical, Gaussian and von Mises distributions would account for a wide range of heterogeneous features, likely increasing the information available to the classifier and its accuracy. Learning and reasoning with these models is not trivial either.

We conclude that using von Mises distributions in Bayesian classifiers, and Bayesian networks generally, is both interesting and challenging. We hope that further research in this area will provide the tools necessary to properly manage directional data in machine learning.

Acknowledgments. This work has been supported by the Spanish Science and Innovation Ministry, Cajal Blue Brain Project (C080020-09), TIN2010-20900-C04-04 and Consolider Ingenio 2010-CSD2007-00018. PL L-C is supported by an FPU Fellowship (AP2009-1772) from the Spanish Education Ministry.

References

1. Fisher, N.I.: *Statistical Analysis of Circular Data*. Cambridge University Press (1993)
2. Mardia, K.V., Jupp, P.E.: *Directional Statistics*. John Wiley and Sons (2000)
3. Minsky, M.: Steps toward artificial intelligence. *Proc. Inst. Radio. Eng.* 49, 8–30 (1961)
4. Duda, R., Hart, P.: *Pattern Classification and Scene Analysis*. John Wiley and Sons (1973)
5. Koller, D., Friedman, N.: *Probabilistic Graphical Models. Principles and Techniques*. The MIT Press (2009)
6. Pourret, O., Naïm, P., Marcot, B.: *Bayesian Networks: A Practical Guide to Applications*. John Wiley and Sons (2008)
7. Domingos, P., Pazzani, M.: Beyond independence: Conditions for the optimality of the simple Bayesian classifier. *Mach. Learn.* 29, 103–130 (1997)
8. Peot, M.A.: Geometric implications of the naive Bayes assumption. In: Horvitz, E., Jensen, F.V. (eds.) *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pp. 414–419. Morgan Kaufman (1996)
9. Perrin, F.: Étude mathématique du mouvement Brownien de rotation. *Ann. Sci. Ec. Norm. Super.* 45, 1–51 (1928)
10. von Mises, R.: Über die “Ganzzahligkeit” der Atomgewichte und verwandte Fragen. *Physikal. Z.* 19, 490–500 (1918)
11. Berens, P.: CircStat: A MATLAB toolbox for circular statistics. *J. Stat. Softw.* 31(10), 1–21 (2009)
12. Kovach, W.L.: Quantitative methods for the study of lycopod megaspore ultrastructure. *Rev. Palaeobot. Palynology* 57(3-4), 233–246 (1989)
13. Berkholz, D.S., Krenesky, P.B., Davidson, J.R., Karplus, P.A.: Protein geometry database: A flexible engine to explore backbone conformations and their relationships to covalent geometry. *Nucleic Acids Res.* 38(suppl.1), D320–D325 (2010)
14. Mardia, K.V., Taylor, C.C., Subramaniam, G.K.: Protein bioinformatics and mixtures of bivariate von Mises distributions for angular data. *Biometrics* 63(2), 505–512 (2007)
15. Boomsma, W., Mardia, K.V., Taylor, C.C., Ferkinghoff-Borg, J., Krogh, A., Hamelryck, T.: A generative, probabilistic model of local protein structure. *Proc. Natl. Acad. Sci. U.S.A.* 105(26), 8932–8937 (2008)
16. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Bajcsy, R. (ed.) *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 1022–1027. Morgan Kaufmann (1993)
17. Bouckaert, R.R.: Estimating replicability of classifier learning experiments. In: Brodley, C.E. (ed.) *Proceedings of the 21st International Conference on Machine Learning*. ACM (2004)
18. Pazzani, M.J.: Searching for dependencies in Bayesian classifiers. *Lecture Notes in Statistics* 112, 239–248 (1995)
19. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. *Mach. Learn.* 29, 131–163 (1997)
20. Sahami, M.: Learning limited dependence Bayesian classifiers. In: Simoudis, E., Han, J., Fayyad, U.M. (eds.) *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 335–338. AAAI Press (1996)
21. Mardia, K.V., El-Atoum, S.A.M.: Bayesian analysis for bivariate von Mises distributions. *J. Appl. Stat.* 37(3), 515–528 (2010)

Unravelling the Yeast Cell Cycle Using the TriGen Algorithm

David Gutiérrez-Avilés, Cristina Rubio-Escudero, and José C. Riquelme

Dpto. Lenguajes y Sistemas Informáticos, Universidad de Sevilla
dgutierrez@alum.us.es, {crubioescudero,riquelme}@us.es

Abstract. Analyzing microarray data represents a computational challenge due to the characteristics of these data. Clustering techniques are widely applied to create groups of genes that exhibit a similar behavior under the conditions tested. Biclustering emerges as an improvement of classical clustering since it relaxes the constraints for grouping allowing genes to be evaluated only under a subset of the conditions and not under all of them. However, this technique is not appropriate for the analysis of temporal microarray data in which the genes are evaluated under certain conditions at several time points. In this paper, we present the results of applying the TriGen algorithm, a genetic algorithm that finds triclusters that take into account the experimental conditions and the time points, to the yeast cell cycle problem, where the goal is to identify all genes whose expression levels are regulated by the cell cycle.

Keywords: microarrays, temporary data, genetic algorithms, yeast.

1 Introduction

The use of high throughput processing techniques has revolutionized the technological research and has exponentially increased the amount of data available [5]. Particularly, microarrays have revolutionized biological research by its ability to monitor changes in RNA concentration in thousands of genes simultaneously [2]. A common practice when analyzing gene expression data is to apply clustering techniques, creating groups of genes that exhibit similar expression patterns. These clusters are interesting because it is considered that genes with similar behavior patterns can be involved in similar regulatory processes [12]. Although in theory there is a big step from correlation to functional similarity of genes, several articles indicate that this relation exists [4]. Traditional clustering algorithms work on the whole space of data dimensions examining each gene in the dataset under all conditions tested. Biclustering techniques [8] go a step further by relaxing the conditions and by allowing assessment only under a subset of the conditions of the experiment, and it has proved to be successful finding gene patterns [6,10]. However, clustering and biclustering are insufficient when analyzing data from microarray experiments where attention is payed on how the time affects gene's behavior. There is a lot of interest in this type of time series experiment because they allow an in-depth analysis of molecular processes in

which the time evolution is important, for example, cell cycles, development at the molecular level or evolution of diseases [1]. Therefore, the use of specific tools for data analysis in which genes are evaluated under certain conditions considering the time factor becomes necessary. The TriGen algorithm goes a step further than clustering and biclustering techniques in the creation of groups of pattern similarity for genes. TriGen works on a three-dimensional space, thus taking into account the time factor, and allowing the evaluation of the behavior of genes only under certain conditions and only under certain time points. TriGen applies an evolutionary technique, genetic algorithms, to find solutions that we refer to as triclusters. We present the results of applying the TriGen algorithm to the yeast cell cycle problem [11], where the objective is to create a comprehensive catalog of yeast genes whose transcript levels vary periodically within the cell cycle. The rest of the paper is structured as follows. Section 2 describes the TriGen algorithm in detail, Section 3 shows the results with synthetic data and with the yeast data. Section 4 summarizes the conclusions reached and proposals for future work.

2 Methodology

In this section we explain the inputs and outputs of the algorithm and we provide a detailed description of the evolutionary process and all the operators implied.

2.1 Input Data

The input data is obtained from temporal microarray experiments. Each of these microarrays reveals the expression level under specific experimental conditions and at an instant of time. Therefore, the input data consists of T number of microarrays, as many as time points to be analyzed. Each value of a microarray for an specific time t represents the level of gene expression of a gene g under a specific experimental condition c .

2.2 Definition of Tricluster

We define a tricluster as a subset of time instants T , a subset of genes G and a subset of conditions C extracted from the input data (described above in section 2.1) which provide a behavior pattern of expression levels of each gene g contained in G under each experimental condition c contained in C and on each time point t contained in T . In comparison of predecessor technologies, a tricluster is, as we said before, a subset of genes, conditions and time points while a cluster is a subset of genes and a bicluster is a subset of genes and conditions. In this particular work, each tricluster contains the expression values of the these three sets and a fitness value that indicates the tricluster's quality. The fitness function will be described in detail in Section 2.3 (Evaluation operator).

2.3 TriGen Algorithm Description

TriGen is based on a genetic algorithm. The evolutionary process is composed for an initialization method in which the initial population will be created with chromosomes or candidate solutions and several operators: evaluation, which measures the quality of each chromosome or individual of the population, selection, which serves to decide which individuals will survive to the next generation, crossover, creates the necessary connections between pairs of individuals to share new genetic material and finally mutation, which performs punctual changes to individuals to ensure genetic variability of future generations, i.e., exploring new spaces of solutions (See Figure 1).

We discuss in detail each of these methods and operators.

```

Input:  Temporary microarray data
Output: Tricluster Solution Set

Begin TriGen algorithm
  Repeat for each Tricluster solution
    Generate Initial Population
    Evaluate population
    Repeat for Number of Generations
      Select Population
      Cross Population
      Mutate Population
      Evaluate Population
    End Repeat
  Select Best One
  Include Best One in Solution Set
End Repeat
End TRIGEN algorithm

```

Fig. 1. TriGen algorithm

Codification of Individuals: Each member of the population represents a tricluster which is a potential solution. It has genetic material that will be manipulated by the genetic operators described in "Genetic Operators" below. This genetic material is composed by a set of chromosomes, they are a subset of time instants T , a subset of genes G and a subset of conditions C extracted from the input data. Each of them is composed by a number of genes, they are the components of the tricluster (they correspond to the components of the input data).

Generation of Initial Population: This method receives a parameter, the number of individuals desired for the initial population. To compose each individual, we choose randomly a subset of timing, genes and conditions of the input data. This process is repeated as many times as specified by the input parameter described above.

Genetic Operators

Selection. A tournament selection mechanism, in which groups of individuals are randomly created sorted from lowest to highest according to the fitness function, and then a random selection from the three groups of the individuals required according to an input parameter is made.

Crossover. This operator completes the population in the next generation P_t generating two new individuals (children) combining the genetic material from two existing ones (parents). For each point of the two parents get two children so the number of crossings is determined by number of individuals who are required to complete the population.

This is a one-point cross that determine a random point cross for the times, genes and conditions and mixing each of the parts to obtain two child by crossing.

Mutation. This operator selects, based on a mutation probability input parameter, a number of individuals who suffer a random out of six: add a time component, a gene component or a condition component, or remove a time component, gene component or condition component.

Evaluation. Since triclustering emerges as an improvement of biclustering to analyze microarray data taking into account the temporal dimension, we have adapted the classical biclustering fitness function, Mean Squared Residue (MSR), presented by Cheng and Church in [3], to the three dimensional space. MSR compares the similarity of each value in the bicluster to the mean values of all genes under the same condition, the mean of the gene under the other conditions included in the bicluster, and the mean of all values in the bicluster. In the case of triclustering, we will assess the similarity of each value not only related to genes and conditions, but also including the temporary plane, i.e., we asses how a gene g behaves under all conditions C at the time points T , how a condition c affects all genes G in time T , and the time factor t in relation to genes G and conditions C , as well as the mean value of all the tricluster. This is formalized as follows:

$$r_{GCT} = \frac{\sum_{g \in G, c \in C, t \in T} r_{gct}^2}{|G| * |C| * |T|} - Weights$$

in the first member of equation, the numerator is:

$$\begin{aligned} f_{gct} = & V_{gct} + M_{GC}(t) + M_{GT}(c) + M_{CT}(g) - \\ & M_G(c, t) - M_C(g, t) - M_T(g, c) - M_{GCT} \end{aligned}$$

where V_{gct} is the tricluster value being evaluated, $M_{GC}(t)$ is the mean of the genes under conditions at a point in time t , $M_{GT}(c)$ is the mean of the genes

over time under a condition c , $M_{CT}(g)$ is the mean of a gene g in time under the conditions, $M_G(c, t)$ is the mean of the genes under one condition and a time point, $M_C(g, t)$ is the mean of the values of a gene at a time point under conditions, $M_T(g, c)$ is the mean of a gene under a condition at all time points and M_{GCT} is the mean value of all points of tricluster.

The denominator factor is:

$$|G| * |C| * |T|$$

where $|G|$, $|C|$ and $|T|$ are, respectively, the number of genes, times and conditions in the tricluster under evaluation.

And the second member of equation, *Weights*, corresponds to:

$$Weights = |G| * w_g + |C| * w_c + |T| * w_t$$

where w_g , w_c and w_t are the weights of the genes, conditions and times for the solution tricluster respectively and $|G|$, $|C|$ and $|T|$ correspond again to the number of genes, times and conditions in the tricluster under evaluation. When increasing the value of one of these weights, we favor the TriGen algorithm finding triclusters with a greater number of components on that term.

3 Results

We show the results obtained applying the TriGen algorithm both to real and to synthetic data.

3.1 Results Using Synthetic Data

Synthetic data are widely used not only for testing the performance of microarray analyzing techniques [7] but also in more general data mining publications [9]. The set of synthetic data has been generated using a software application developed for such purpose. For this particular work, we have simulated data from 5 different time points and 10 conditions using microarrays containing 1000 genes. Each gene is assigned a random value which is contained in the rank, respectively for each condition, [1, 15], [7, 35], [60, 75], [0, 25], [30, 100], [71, 135], [160, 375], [5, 30], [25, 40] y [10, 30]. In such data set, we have allocated a tricluster with all its values fixed to 1. The size of the tricluster is *time* = 5, *genes* = 8 and *conditions* = 8. TriGen was able to successfully find a solution containing the aforementioned tricluster. The execution was made with the following parameters: 100 generations and 500 members in the population. The selection parameter is 70% and the mutation probability is 5%. The weight values have been adjusted to $w_g = 0.01$, $w_c = 0.55$ y $w_t = 0.35$, in order to favor the number of conditions and time points, since the genes show high dimensionality in relation to conditions and time.

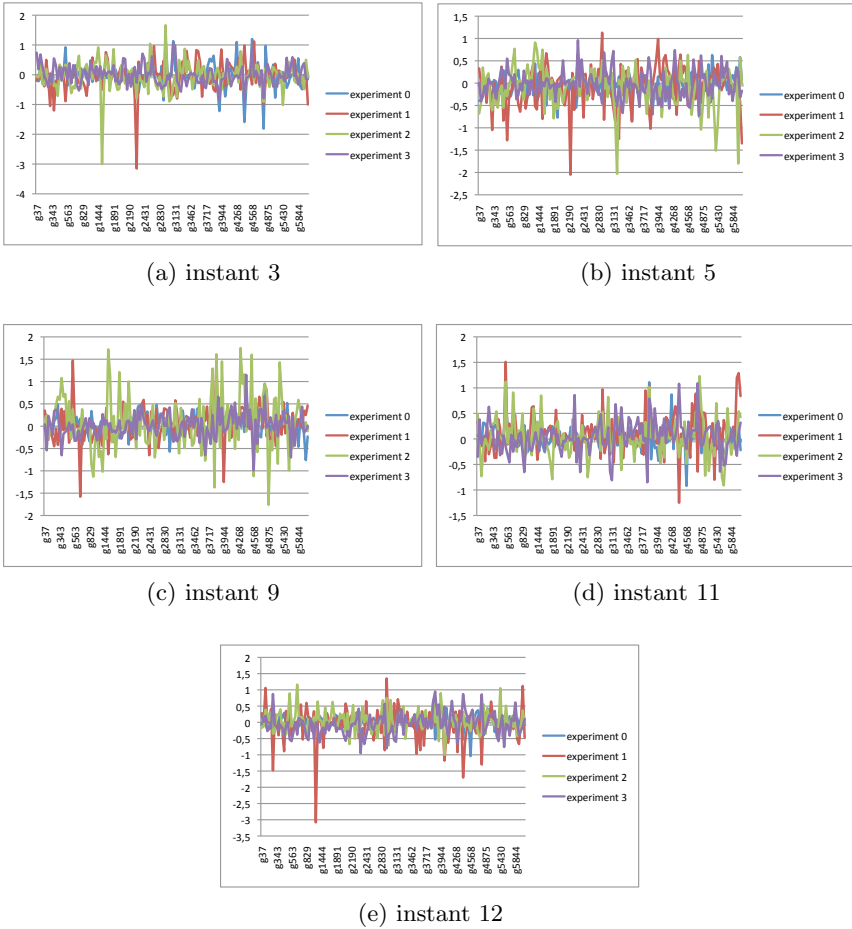


Fig. 2. Gene expression values under four experiments at instant 3 (a), 5 (b), 9 (c), 11 (d) and 12 (e)

3.2 Results Using Real Data

We have applied the TriGen algorithm to the yeast (*Saccharomyces Cerevisiae*) cell cycle problem [11]. The yeast cell cycle analysis project's goal is to identify all genes whose mRNA levels are regulated by the cell cycle. By applying TriGen to this dataset, we aim to find a pattern on this cell cycle.

The data is available in <http://genome-www.stanford.edu/cellcycle/>. In this experiment, 6179 genes are analyzed under 6 conditions, termed *cln3*, *clb2*, pheromone, *cdc15*, *cdc28* and elutriation [11]. Samples were taken at 2 time points for *cln3*, 2 for *clb2*, 18 for pheromone, 24 for *cdc15*, 17 for *cdc28* and 14 for elutriation. To apply the TriGen algorithm we did not take into account the conditions with only 2 time points, since they are not relevant for a time course

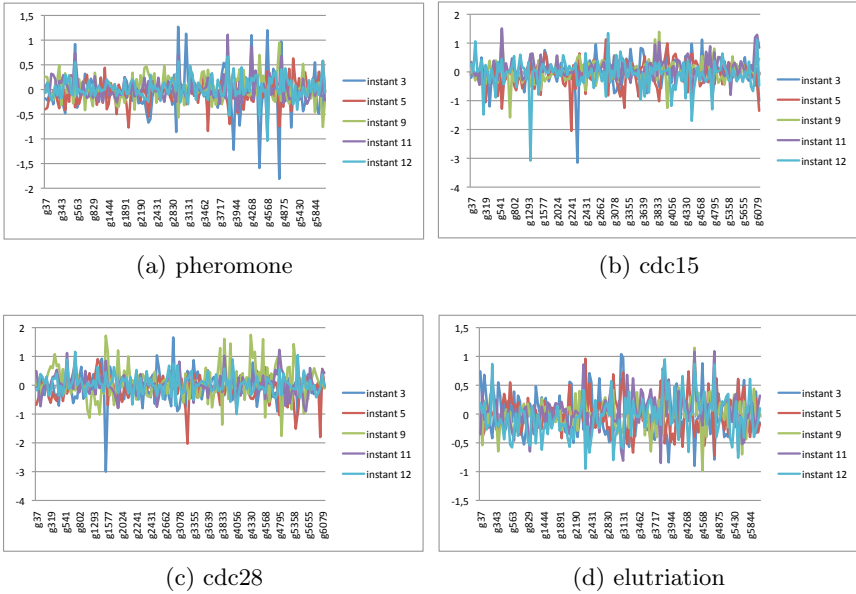


Fig. 3. Gene expression values under five instants at pheromone (a) cdc15 (b) cdc28 (c) and elutriation (d) experiments

experiment, so we used the first 14th time points of the pheromone, cdc15, cdc28 and elutriation experiment. Therefore our dataset contains 14 time points, 6179 genes and 4 conditions. The algorithm has been executed to extract 10 solutions, i.e. 10 triclusters with the following parameters: 100 generations, 50 members in the population, 50% for selection probability and 70% for mutation. The weights applied have been $w_g = 0.0$, $w_c = 100.0$ y $w_t = 0.0002$, thus we favored the condition dimension and penalized gene dimension to get a reduced subset of genes on solution triclusters.

For legibility reasons we focuss in one of the solutions, a tricluster gathering 142 genes under the 4 experiments, pheromone (experiment 0), cdc15 (experiment 1), cdc28 (experiment 2) and elutriation (experiment 3), and 5 time points, instants 3, 5, 9, 11 and 12.

We show three groups of graphics related to this solution: In Figure 2 we present the outline of the gene expression values (Y axis) for each solution gene point (X axis) comparing the pheromone (experiment 0), cdc15 (experiment 1), cdc28 (experiment 2) and elutriation (experiment 3) experiments setting time points to instants 3 (a), 5 (b), 9 (c), 11 (d) and , 12 (e). In Figure 3 we present the outline of Gene expression values (Y axis) for each solution gene point (X axis) comparing 3, 5, 9, 11 and 12 time points setting the experiments to pheromone (a) cdc15 (b) cdc28 (c) and elutriation (d). Finally in Figure 4 we present the outline of Gene expression values (Y axis) for each time point (X axis) comparing each solution gene setting the experiments to pheromone (a) cdc15 (b) cdc28 (c) and elutriation (d).

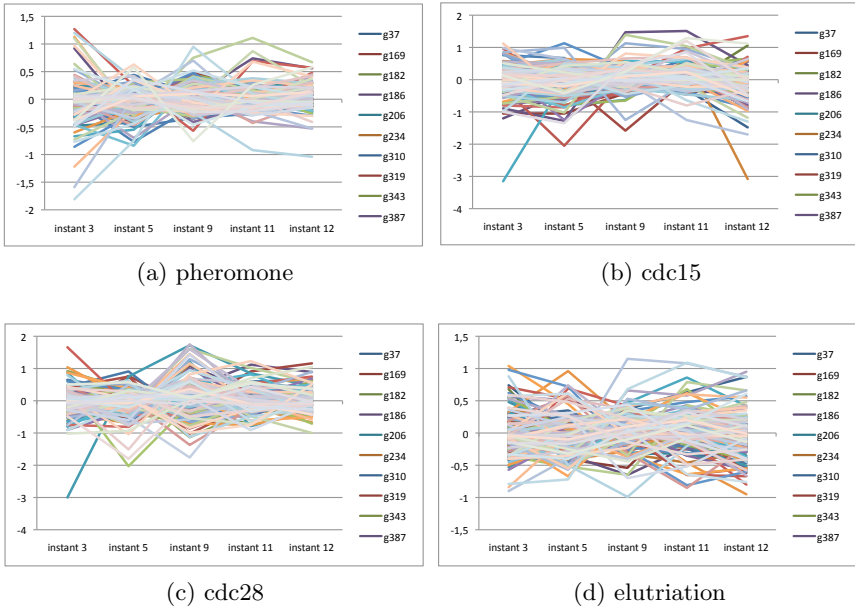


Fig. 4. Gene expression for five instants under gene solution set at pheromone (a) cdc15 (b) cdc28 (c) and elutriation (d) experiments

We see that the algorithm has been capable to group together genes with very similar gene expression values (comprised in the $[-2, 2]$ interval) for the three dimensions visited. Therefore, TriGen has shown its ability to mine groups of co-expressed genes taking into account the time dimension.

4 Conclusions and Future Work

We have presented the results obtained by applying the tricluster algorithm TriGen to the yeast cell cycle problem. TriGen represents a step further than clustering and biclustering in the analysis of temporal microarray data since it groups genes which exhibit a similar behavior under a subset of conditions and under a subset of time points. It is genetic based algorithm, with an evaluation function developed as the natural 3D extension from the classic function evaluation for biclustering proposed by Cheng y Church in [3]. The results show that the algorithm is capable to mine triclusters of genes based on their expression levels. TriGen is still in an early development stage, so there is still a lot of work to do, not only for the algorithm, such as a deeper study of the evaluation function or parallelization of the algorithm to make it faster, but also for the validation phase or the application of this algorithm for other types of data, such as image analysis.

References

1. Bar-Joseph, Z.: Analyzing time series gene expression data. *Bioinformatics* 20(16), 2493 (2004)
2. Brown, P., Botstein, D.: Exploring the new world of the genome with dna microarrays. *Nature Genet.* 21(suppl.), 33–37 (1999)
3. Cheng, Y., Church, G.: Biclustering of expression data. In: Proceedings/.. International Conference on Intelligent Systems for Molecular Biology; ISMB. International Conference on Intelligent Systems for Molecular Biology, vol. 8, p. 93 (2000)
4. D’haeseleer, P., Liang, S., Somogyi, R.: Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics* 16(8), 707–726 (2000)
5. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press (1998)
6. Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* 95(25), 14863 (1998)
7. Hakamada, K., Okamoto, M., Hanai, T.: Novel technique for preprocessing high dimensional time-course data from dna microarray: mathematical model-based clustering. *Bioinformatics* 22(7), 843 (2006)
8. Hartigan, J.A.: Direct clustering of a data matrix. *Journal of the American Statistical Association* 67(337), 123–129 (1972)
9. Pargas, R., Harrold, M., Peck, R.: Test-data generation using genetic algorithms. *Software Testing Verification and Reliability* 9(4), 263–282 (1999)
10. Pontes, B., Divina, F., Giráldez, R., Aguilar-Ruiz, J.: Improved biclustering on expression data through overlapping control. *International Journal of Intelligent Computing and Cybernetics* 3(2), 293–309 (2010)
11. Spellman, P., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell* 9(3), 3273–3297 (1998)
12. Tan, M., Smith, E., Broach, J., Floudas, C.: Microarray data mining: A novel optimization-based approach to uncover biologically coherent structures. *BMC Bioinformatics* (in press)

Pattern Recognition in Biological Time Series

Francisco Gómez-Vela, Francisco Martínez-Álvarez, Carlos D. Barranco,
Norberto Díaz-Díaz, Domingo Savio Rodríguez-Baena,
and Jesús S. Aguilar-Ruiz

Department of Computer Science, Pablo de Olavide University of Seville
`{fgomez, fmaralv, cdbargon, ndiaz, dsrodbae, jaguilar}@upo.es`

Abstract. Knowledge extraction from gene expression data has been one of the main challenges in the bioinformatics field during the last few years. In this context, a particular kind of data, data retrieved in a temporal basis (also known as time series), provide information about the way a gene can be expressed during time. This work presents an exhaustive analysis of last proposals in this area, particularly focusing on those proposals using non-supervised machine learning techniques (i.e. clustering, biclustering and regulatory networks) to find relevant patterns in gene expression.

Keywords: Clustering, biclustering, genetic regulatory networks, time series, gene expression data.

1 Introduction

Collecting data along the time is a very common task in a wide variety of fields, as engineering, medicine, or bioinformatics. In this context, the concept of a *time series* as a sequence of values measured along with the time, and therefore chronologically ordered, is defined. Even though time is a continuous variable, equidistance in time measures is used in practise.

Having knowledge on the past behavior of a variable can be very valuable if we want to predict its future behavior. Moreover, studying of other variables related to the subject variable can contribute with additional information for creating a model of the process.

In the particular case of time series of gene expression data, the length of the data series is usually short. It is usual to have only a few samples of data to work with for extracting data patterns. That means that classical machine learning techniques must be adapted for this particular situation. The recent increment of interest on discovering gene behavior patterns has motivated an increment of relevant works on the context of clustering.

Additionally, the powerful microarray techniques mean a great improvement in bioinformatics, as they make possible to simultaneously analyze the behavior of thousands of genes under different conditions (usually tens of them). As a result of using this technology, we obtain a gene expression database. This is a set of real numbers representing the quantity of mRNA (messenger RNA)

resulting from the expression of a gene under a particular condition. Machine learning techniques are usually used to correctly interpret this amount of data, including supervised learning (i.e. neural networks, support vector machines...) and unsupervised learning (i.e. clustering, biclustering...).

This work focuses on surveying the last proposals based on applying clustering, biclustering and regulatory networks (RN) to time series of gene expression data. Particularly, we will consider those techniques specifically aimed to use data from experiments where the different experimental conditions are different time moments of an evolving biological process. To sum up, we focus on techniques to analyze a sequence of gene expression samples, strategically selected along the time, to predict the future gene behavior.

The rest of the paper is organized as follows. Sections 2 and 3 are devoted to describe the different techniques considered in this work: clustering and biclustering (Sect. 2) and RNs (Sect. 3). An exhaustive, up to the date, listing of (bi)clustering and RNs proposals applied to time series of gene expression data can be found in Section 4. Finally, Section 5 includes a discussion on these proposals and the potential of their results and some concluding remarks.

2 Classification Techniques: Clustering and Biclustering

Forecasting and describing data are two of the main goals of data mining. In the context of data description, classification techniques help organizing a data set by grouping data elements in classes. Among non-supervised classification techniques, *clustering* and *biclustering* highlight. The goal of these techniques is to sort data in groups, which are named (bi)clusters, where data items belonging to a group share some common characteristics and similarities that makes them different from other groups of data items. Note that unsupervised techniques mean that there are no predefined classes nor classifications examples used to do the task.

2.1 Clustering Techniques

The main goal on applying clustering techniques [26] on gene expression data in a *microarray* is to group all these genes whose expression level behaves in a similar way along all tested experimental conditions. This way, the genes belonging to the same cluster are said to be *co-expressing*. This is a very useful information to find cooperating genes in a chemical *pathway* and a sign that these genes are involved in a particular biological process. Discovering this gene relation is critical for obtaining a global vision of cell activity. Given this situation, detecting genes behaving in a similar way is a very important factor, however it is not the only important factor in this task (i.e. the interaction between these gene byproducts [24]).

Given the nature of gene expression data and the biological focus of the research made on it, applying clustering techniques on gene expression data is a very specific problem. Clustering is the first step in data mining and knowledge

discovery. The aim, in this case, to define clusters is to discover the natural structure and distribution of gene expression data and, for this task, there is usually no previous knowledge to help. Therefore, a good clustering algorithm candidate should not depend on previous knowledge.

A clustering algorithm capable of estimating the *real* number of clusters in a database (as, for instance, Expectation-Maximization algorithm) is more suited for application in clustering gene expression data than other that requires setting a predefined cluster number (i.e. k-means algorithm). Thus, requiring user parameters in this context is problematic. The greater the number of parameters is, the more dependent of the parameters values combination the result is, making complex a right selection of parameter values. Regarding data input, microarray data is usually very noisy due to the complex process to obtain it.

Due to the aforementioned reasons, in spite of the great noise level in data, a clustering algorithm applied to gene expression matrix analysis must be able to extract useful knowledge. Besides, the data normally refers to a high number of attributes (genes). Given the size of these gene expression matrices, it is necessary to reduce the computing time and required hardware resources. In addition, it is worth highlighting that recent empirical studies have shown that gene expression data is very self-related [10] and that an overlap between clusters can be possible. This can result in cluster intersections or clusters inside clusters [22] and cluster algorithms must take this into account. Finally, in [23] clustering techniques are applied (particularly k-means) to discover behavior patterns in gene expression data taken along time. This work also proposes an multi-objective optimization algorithm to associate each genetic pattern with the techniques best suited to find genes following the pattern.

2.2 Biclustering Techniques

Applying clustering algorithms on gene expression data usually does not provide the best results. Much of the activity patterns of gene groups are only present under a particular set of experimental conditions. Actually, the available knowledge on cell processes suggests that, while a subset of genes is co-regulated and co-expressed under particular experimental conditions, under different conditions these genes can show independent behavior. Discovering this local behavior patterns can be the key to discover gene pathways, which could be hard to discover in other ways. For this reason, the paradigm of clustering techniques must change to methods that allow local pattern discovery in gene expression data [1].

Clustering techniques can only be separately applied on the rows or the columns of a gene expression matrix. However, *biclustering* techniques [13] can detect clusters in a bidimensional context, considering row and columns simultaneously. This means that clustering techniques discover only global models whereas biclustering techniques can discover local models. When a clustering algorithm is applied, each gene belonging a cluster is considered on all experimental conditions. The same way, each condition in a cluster is considered by analyzing all genes in the matrix. When biclustering techniques are applied, each gene is selected for being part of a bicluster only by its behavior under a subset

of conditions and to select each condition on a bicluster only a subset of genes is taken into account. Concluding, the aim of biclustering techniques is to identify subgroups of genes and conditions applying a simultaneous clustering process on rows and columns, instead of making independent analyses of each one of the two dimensions. Biclustering methods are ideal when the following situations can be found on data:

1. Only a small group of genes participates in an interesting cellular process.
2. An interesting cellular process is shown under a subset of experimental conditions.
3. Only one gene is participating in several pathways that are not shown in every experimental condition.

The concept of bicluster defines a more flexible computational framework as there are less restrictions: the submatrixes do not have to be exclusive nor inclusive, that is, a gene or a condition can belong to a bicluster, to more than one or to none. Hence, the lack of structural rules for biclusters means degrees of freedom, reducing the high probability of getting too much overlapping results. Because of this, biclustering techniques must guarantee the quality and significance of the results. Therefore, an additional aim of these techniques is to provide some quality measures of the results (as statistical models, heuristic scores or biological analysis of the results) to guarantee the significance of the found biclusters.

3 Functional Analysis: Gene Regulatory Networks

In recent years, much research effort has been done to develop methodologies to generate gene networks (GN). This increasing interest is because GNs let discover the existing dependencies and interactions among genes. They are a very intuitive visual solution to gene-gene interaction and they are even able to completely model a particular biological process.

GNs are usually represented as graphs where nodes symbolize genes and edges relations, reactions or interactions between them. There are different kinds of GNs depending on the network generation method used and the information providing the network itself. This way, the edges in the network can be directed or undirected and weighted or not. A GNs is *directed* when there is an origin and destination in the gene relations, which is represented by the direction of the edges in the graph. On the other hand, in an *undirected* GN, it is not possible to represent the interactions directions. In the case of *weighted* GNs (or *labeled* GNs), the labels of edges can represent different factors, depending on the information that want to be represented in each case. For instance, in bayesian networks, the labels represent the probability on which the regulation represented by the edge can take place.

When inferring GNs, it is necessary to choose well the architecture followed to generate it. Selecting the architecture on which our methodology generates GNs is crucial for, among other things, the network morphology. A bad choice

can induce poorly relevant or even wrong results, therefore all possibilities must be carefully considered. Regarding the architecture, there are a great number of network types. Among them, the following are remarkable [7]: information theory based methods, boolean networks, differential equation based networks, bayesian networks and, the more recent, tree models [20].

Inside the category of GNs based on information theory, the networks obtained by using correlation measures (correlation networks) are the most notable. This kind of networks are usually represented as undirected graphs with edges whose label is the correlation coefficient between the expression level of the genes. Boolean GNs are the group of networks that provide least information, as the state of genes is binary (expressed or unexpressed). However, this group has been recently combined with others to enrich the final result. Differential equations based GNs are very useful to model complex relations between genes, as the gene expression changes are modeled by an inferred differential equation. Bayesian networks are, possibly, the most known group due to the existing number of works on them. They use Bayes probability theory to model the networks. Finally, tree-model based networks, which are a derivation of regression trees, have recently been introduced as a new alternative to generate gene networks.

4 Unsupervised Machine Learning on Gene Expression Temporal Series Data: An Overview

4.1 Clustering Based Proposals

Based on the concept of temporality, Jaqaman et al. [9] developed a clustering technique to group yeast strains from high resolution dynamic measuring of living cell chromosomes.

A short time after, Chang-Tsun et al. [12] tackled the problem of gene expression time series clustering by developing probabilistic models that were stable and accurate. The authors stated that previous models, up to that time, were computationally prohibitive unless some independence assumptions are made to describe large-scale data. To overcome the above limitation, they introduced an unsupervised model based on conditional random fields.

At the end of 2009, Magni et al. [16] proposed one of the first tools to cluster gene expression time series. The tool, named TimeClust, includes two new algorithms based on *hierarchical clustering* and *self-organizing maps*. The first approach is an agglomerative clustering technique based on cluster distance measures and a linking method. The above approach is fully customizable, users can choose the linking method (simple, complete, average, centroid), the number of clusters and the number of nodes shown in the resulting dendrogram. The second approach is based on a cluster representation using a stochastic population model. Both approaches use ordinary clustering techniques by transforming the time series in time intervals where the attribute values increase, decrease or keep constant.

The same year, Kiddle et al. [3] conducted a research on gene expression data including time delays, inversions and transient correlations. To do so, they

extended the *Affinity Propagation* algorithm [4] considering the temporal nature of data.

Recently, Krishna et al. [11] proposed a Granger Causality [5] based on clustering technique. This measure, previously introduced in bioinformatics for reverse engineering of microarray data [6,18,19], provides a method to evaluate the influence of one time series on another. In particular, Krishna et al. made use of Granger Causality in combination with a theoretic-graphical methodology to build association matrixes between genes and, thus, detect functional modules in data. In their experiment description, they stated that their proposal was very simple to implement as well as statistically traceable, able to produce sets of functionally related genes that can be used by gene circuit inverse engineering techniques.

4.2 Biclustering Based Proposals

Although the classic biclustering is able to discover co-regulated genes under particular conditions, it ignores the inherently sequential relationship between samples taken in different time stamps, meaning that it is unable to analyze gene expression time series. However, Zhang et al. [27] propose to apply a biclustering algorithm to this kind of data by using the mean squared residue score as measure. After selecting a threshold for the measure, the algorithm alternatively deletes genes and samples, according to their correlation to the bicluster. Besides, only deleting points from the beginning or the end of a time interval is allowed, so that the simultaneous maximization of the number of genes and the length of the time interval is guaranteed.

One of the most referenced works on biclustering time series is the one from Sara Madeira and Arlindo Oliveira in 2005 [14] and the later extended version in 2009 [15]. The *e-CCC-Biclustering* biclustering algorithm is able to discover all the maximal coherent biclusters in contiguous columns of a temporal gene expression database, with a polynomial complexity on the database size. By using efficient techniques of string processing based on suffix trees, this algorithm works on a discretized version of the original matrix. A *ccc-bicluster* (contiguous column coherent bicluster) $A_{I,J} = (I; J)$ is composed by the subset $I = i_1 \dots i_k$ and the subset of contiguous columns $J = r, r + 1, \dots, s - 1, s$ of the A matrix, such that $A_{ij} = A_{lj}$ for every $i, l \in I$ and $j \in J$. Each bicluster represents a behavior pattern according to the gene expression. The last versions of this algorithm include very significant improvements as, for instance, the ability to deal with null values, to discover expression patterns with any kind of correlation, as long as different ways to compute a particular allowed error degree. Finally, authors propose a statistical method to measure the quality of each bicluster, by combining the statistical relevance of each pattern with the similarity measure of the overlapping clusters.

One of the latest proposals we can find in literature is to adapt a fuzzy set theory based clustering algorithm for biclusters discovery, constant or based on a similarity measure [21] as well. Particularly, this proposal makes use of *Fuzzy C-Means* [2]. This algorithm divides the data space in fuzzy partitions through

an iterative optimization process of an objective function. Starting from this algorithm, authors implement two models that are suited for two dimensional clustering. The first one aims to detect constant biclusters, where each bicluster centroid is a particular value. Besides, a penalty parameter is included to guarantee a consecutive temporal basis of the results. In the second model the distance measure is changed to obtain coherent biclusters. A particular problem of this technique is that it is required to set the number of clusters that must be discovered.

4.3 Regulatory Networks Proposals

A good proposal on gene RNs is the one in [17]. It introduces a method for generating RNs from temporal microarray data. The first step in the methodology is to group and discretize gene expression data by using k-means and support vector regression. After that, the boolean activation–inhibition networks are enumerated and matched to the discretized data. The most novel contribution of this work is the use of a dynamic model (bayesian networks) combined with the boolean models to generate networks.

In [28], Zhao et al. propose an algorithm to discover gene networks, by using the *minimum description length* principle, that significantly reduces the search space for graphical solutions and achieves a good balance between model complexity and data fitting. The results obtained during the experiments on synthetic networks show good performance of the algorithm. After comparing the proposal to other techniques in the literature it can be concluded that the algorithm stands out due to its efficiency, precision, robustness and scalability, being the above their most interesting features.

The relation between clustering and network generation techniques is becoming narrower by the time. Clustering genes on their behavior patterns let the functional study of the genes belonging to each cluster. Once this study is done, it is relatively easy to discover gene–gene relations between genes from the same or different clusters. Relative to the above proposal, in 2010 a work on network generation based on temporal expression patterns was proposed [8]. This new method, named NACEP, compares the temporal patterns of a gene between two different experimental conditions, taking into account all possible co-expression modules in which this gene participates. As a first step, a clustering process on all genes is performed by only considering two selected experimental conditions. In the second step, the temporal pattern of a gene is inferred as a weighted mean of the temporal patterns of all obtained clusters, where the weight of each cluster is the probability of membership of the gene to it. Finally, the temporal patterns of each gene between two experimental conditions are compared to a non-parametrical test.

Other related work on clustering and network generation is [25]. It introduces a new statistical method for inferring gene networks based on clustering. This algorithm is able to predict, at the same time, temporal expression profile clusters, relations between clusters and relations between clusters and stimulations.

5 Conclusion

This work has shown an overview on the different techniques, up to now, to extract information from gene expression data. This study is particularly focused on techniques able to work on data sorted as time series.

Particularly, clustering, biclustering and regulatory networks proposals are introduced, proving the usefulness of these techniques to solve particular problems in this research area. However, these proposals are still particular solutions, meaning that most of them, even though they are innovative, are becoming obsolete as the problems evolve.

Acknowledgements. Authors want to thank financial support from “Junta de Andalucía”, P07-TIC-02611 project, and “Ministerio de Ciencia e Innovación”, TIN2007-68084-C-02 project.

References

1. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. In: Proceedings of the 6th International Conference on Computational Biology, pp. 49–57 (2002)
2. Bezdek, J.C.: Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York (1981)
3. Kiddle, S.J., et al.: Temporal clustering by affinity propagation reveals transcriptional modules in *Arabidopsis thaliana*. *Bioinformatics* 26(3), 355–362 (2010)
4. Frey, B.J.J., Dueck, D.: Clustering by passing messages between data points. *Science* 315(5814), 972–976 (2007)
5. Granger, C.W.J.: Investigating causal relations by econometric models and cross-spectral methods. *Econometrica* 37(3), 424–438 (1969)
6. Guo, S., Wu, J., Ding, M., Feng, J.: Uncovering Interactions in the Frequency Domain. *PLoS Computational Biology* 4(5), e1000087+ (2008)
7. Hecker, M., Lambeck, S., Toepfer, S., van Someren, E., Guthke, R.: Gene regulatory network inference: Data integration in dynamic models – a review. *Biosystems* 96(1), 86103 (2009)
8. Huang, W., Cao, X., Zhong, S.: Network-based comparison of temporal gene expression patterns. *Bioinformatics* 26(23), 2944–2951 (2010)
9. Jaqaman, K., Dorn, J.F., Marco, E., Sorger, P.K., Danuser, G.: Phenotypic clustering of yeast mutants based on kinetochore microtubule dynamics. *Bioinformatics* 23(13), 1666–1673 (2007)
10. Jiang, D., Pei, J., Zhang, A.: Interactive exploration of coherent patterns in time-series gene expression data. In: Proceedings of SIGKDD (2003)
11. Krishna, R., Li, C.T., Wollaston, V.B.: A temporal precedence based clustering method for gene expression microarray data. *BMC Bioinformatics* 11(1), 68+ (2010)
12. Li, C.T., Yuan, Y., Wilson, R.: An unsupervised conditional random fields approach for clustering gene expression time series. *Bioinformatics* 24(21), 2467–2473 (2008)
13. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics* 1(1), 24–45 (2004)

14. Madeira, S.C., Oliveira, A.L.: A linear time biclustering algorithm for time series gene expression data. Technical Report: INESC-ID, pp. 1–8 (2005)
15. Madeira, S.C., Oliveira, A.L.: A polynomial time biclustering algorithm for finding approximate expression patterns in gene expression time series. *Algorithms for Molecular Biology* 4(8), 1–39 (2009)
16. Magni, P., Ferrazzi, F., Sacchi, L., Bellazzi, R.: TimeClust: a clustering tool for gene expression time series. *Bioinformatics* 24(3), 430–432 (2008)
17. Martin, S., Zhang, Z., Martino, A., Faulon, J.L.: Boolean dynamics of genetic regulatory networks inferred from microarray time series data. *Bioinformatics* 23(7), 866–874 (2007)
18. Mukhopadhyay, N.D., Chatterjee, S.: Causality and pathway search in microarray time series experiment. *Bioinformatics* 23(4), 442–449 (2007)
19. Nagarajan, R., Upreti, M.: Comment on causality and pathway search in microarray time series experiment. *Bioinformatics* 24(7), 1029–1032 (2008)
20. Nepomuceno-Chamorro, I.A., Aguilar-Ruiz, J.S., Riquelme, J.C.: Inferring gene regression networks with model trees. *BMC Bioinformatics* 11, 517 (2010)
21. Qu, J., Ng, M., Chen, A.L.: Constrained subspace clustering for time series gene expression data. In: *The Fourth International Conference on Computational Systems Biology*, pp. 323–330 (2010)
22. Rubio-Escudero, C., Martínez-Álvarez, F., Romero-Zaliz, R., Zwir, I.: Classification of gene expression profiles: Comparison of K-means and Expectation-Maximization algorithms. In: *Proceedings of the 8th International Conference on Hybrid Intelligent Systems*, pp. 831–836 (2008)
23. Rubio-Escudero, C., Romero-Zaliz, R., Zwir, I., del Val, C.: Optimization of multi-classifiers for computational biology: application to gene finding and expression. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling* 125(3), 599–611 (2010)
24. Segal, E., Wang, H., Koller, D.: Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics* 19(2), 264–272 (2003)
25. Shiraishi, Y., Kimura, S., Okada, M.: Inferring cluster-based networks from differently stimulated multiple time-course gene expression data. *Bioinformatics* 26(8), 1073–1081 (2010)
26. Xu, R., Wunsch II, D.C.: Survey of clustering algorithms. *IEEE Transactions on Neural Networks* 16(3), 645–678 (2005)
27. Zhang, Y., Zha, H., Chu, C.H.: A time-series biclustering algorithm for revealing co-regulated genes. *Bioinformatics* 18(3), 606–611 (2005)
28. Zhao, W., Serpedin, E., Dougherty, E.R.: Inferring gene regulatory networks from time series data using the minimum description length principle. *Bioinformatics* 22(17), 2129–2135 (2006)

On the Expressive Power of First Order-Logic Extended with Allen’s Relations in the Strict Case

Willem Conradie¹ and Guido Sciavicco²

¹ Department of Mathematics, University of Johannesburg,
Johannesburg, South Africa
wconradie@uj.ac.za

² Department of Information, Engineering and Communications,
University of Murcia, Murcia, Spain
guido@um.es

Abstract. We consider the languages of first order-logic (with equality) extended with Allen’s relations for temporal intervals. We give a complete classification of such languages in terms of relative expressive power, thus determining how many, and which, are the intrinsically different extensions of first-order logic with one or more of Allen’s relations. Classifications are obtained for three different classes of interval structures, namely those based on arbitrary, discrete, and dense linear orders. The strict semantics (where point-intervals are excluded) is assumed throughout.

1 Introduction

The relevance of interval temporal logics in many theoretical and applied areas of computer science and AI, such as theories of action and change, natural language analysis and processing, and constraint satisfaction problems, is widely recognized. Interval temporal logics formalize reasoning about interval structures over ordered domains, where time intervals, rather than time instants, are the primitive ontological entities. The variety of binary relations between intervals in linear orders was first studied systematically by Allen [All83] (see Tab. 1), who explored their use in systems for time management and planning.

While in the recent literature researchers have focused on modal logics for (interval) temporal reasoning (see, e.g., the survey article [GMG04]), in this paper we are interested in the first-order framework on which the modal languages are based. As presented in the early work of Allen and Hayes [AH85] and van Benthem [vB83], temporal reasoning over intervals can be formalized as an extension of first-order logic with equality with one or more interval relations, and the properties of the resulting language can be studied. In this paper we ask the question: how many and which expressively different languages can be obtained by enriching first-order logic with combinations of Allen’s interval relations? Since there are 12 Allen relations (excluding equality), 2^{12} is an upper bound

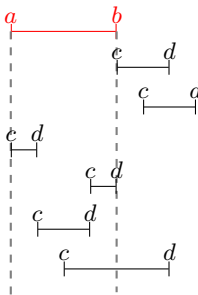
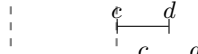
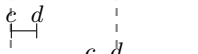
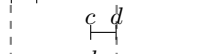
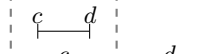

to this number. However, since certain relations are definable in terms of other ones, the actual number is less and in fact, as we will show, much less. The answer will also depend on our choices of certain semantic parameters, specifically, the class of linear orders over which we construct our interval structures.

In this paper we restrict ourselves to the so-called strict semantics, where point-intervals are omitted, and we consider the classification problem over three different classes of linear orders. Apart from the intrinsic interest and naturalness of this classification problem, its outcome has some important repercussions, principally in the reduction of the number of cases that need to be considered in other problems relating to these languages. For example, it reduces the number of representation theorems that are needed: given the *dual* nature of time intervals (i.e., they can be abstract first-order individuals with specific characteristics, or they can be defined as ordered pairs over a linear order), one of the most important problems that arises is the existence or not of a *representation theorem*. Consider any class of linear orders: given a specific extension of first order logic with a set of interval relations (such as, for example, *meets* and *during*), does there exist a set of axioms in this language which would constrain (abstract) models of this signature to be isomorphic to concrete ones? In other words can we produce an isomorphism to models in which the domain is the set of intervals over the considered linear order, and in which the relations are the concrete interval relations? In the relevant literature, we find a number of representation theorems: van Benthem [vB83], over rationals and with the interval relations *during* and *before*, Allen and Hayes [AH85], for the dense unbounded case without point intervals and for the relation *meets*, Ladkin [Lad78], for point-based structures with a quaternary relation that encodes meeting of two intervals, Venema [Ven90], for structures with the relations *starts* and *finishes*, Goranko, Montanari, and Sciavicco [GMG03], that generalizes the results for structures with *meets* and *met-by*, and Coetzee [Coe09] for dense structure with *overlaps* and *meets*. Clearly, if two sets of interval relations give rise to expressively equivalent languages, two separate representation theorems for them are not needed. In which cases are representation theorems still outstanding? In this paper we will show that there are exactly 10 different such extensions of first-order logic (with equality) when interpreted over the class of all linear orders or of all discrete linear orders, and 4 when interpreted over the class of dense linear orders. Due to the lack of space, most results are just sketched.

2 First Order Logic and Allen's Relations

Let us consider a linearly ordered set $\mathbb{D} = \langle D, < \rangle$. In the *strict* semantics, in which we are interested here, an *interval* over a linear order is defined as an ordered pair $[a, b]$ such that $a < b$. The *non-strict* semantics only requires that $a \leq b$, thus including degenerate objects of the type $[a, a]$. The set of all intervals on \mathbb{D} is denoted by $\mathbb{I}(\mathbb{D})$. The variety of all possible binary relations between intervals has been studied by Allen [All83]. There are thirteen Allen's relations, including equality. Table 1 gives and illustrates the definitions of 6 of these

Table 1. Allen’s interval relations, excluding equality

$[a, b] m [c, d] \Leftrightarrow b = c$	
$[a, b] b [c, d] \Leftrightarrow b < c$	
$[c, d] s [a, b] \Leftrightarrow a = c, d < b$	
$[c, d] f [a, b] \Leftrightarrow b = d, a < c$	
$[c, d] d [a, b] \Leftrightarrow a < c, d < b$	
$[a, b] o [c, d] \Leftrightarrow a < c < b < d$	

relations, the other 7 consisting of the inverses of those illustrated and equality (which is of course equal to its own inverse). For each relation r , its inverse is denoted by ri . Since we will always be assuming equality in our language, we will only need to deal explicitly with the other twelve relations. We denote this set by $AL = \{m, b, s, f, d, o, mi, bi, si, fi, di, oi\}$.

Given a subset $S = \{r_1, \dots, r_n\} \subseteq AL$ of Allen’s relations, a *concrete interval structure of signature S* is a relational structure $\mathcal{I} = \langle \mathbb{I}(\mathbb{D}), r_1, r_2, \dots, r_n \rangle$, where each r_i is defined on $\mathbb{I}(\mathbb{D})$ according to Tab. 1. \mathcal{I} is further said to be *of the class C* when \mathbb{D} belongs to the specific class of linearly ordered sets C . Since all thirteen of Allen’s relations are already implicit in $\mathbb{I}(\mathbb{D})$, we will often simply write $\langle \mathbb{D}, \mathbb{I}(\mathbb{D}) \rangle$ for a concrete interval structure $\langle \mathbb{I}(\mathbb{D}), r_1, r_2, \dots, r_n \rangle$. This is in accordance with the standard usage in much of the literature on interval temporal logics. We denote by $FO + S$ the language of first-order logic with equality and relation symbols corresponding to the relations in S .

Definition 1. Let $S \subseteq AL$ and C a class of linear orders. We say that $FO + S$ defines $r \in AL$ over C , denoted by $FO + S \rightarrow_C r$, if there exists $FO + S$ -formula $\varphi(x, y)$ such that $\varphi(x, y) \leftrightarrow r(x, y)$ is valid on the class of concrete interval structures of signature $(S \cup \{r\})$ based on C . Note that $FO + S \rightarrow_C r$ for all $r \in S$.

When $FO + S$ defines r over the class of all linear orders, then we simply write $FO + S \rightarrow r$. Finally, notice that for each $r \in AL$, $FO + \{r\} \rightarrow ri$; indeed, we have that $ri(x, y) \leftrightarrow r(y, x)$. Moreover, because we are working with languages with equality, $FO + \emptyset \rightarrow =$. Therefore, we can limit our attention to the set $AL^+ = \{m, b, s, f, d, o\}$.

Definition 2. Let $S \subseteq AL^+$ and C a class of linear orders. We say that S is complete over C if and only if $FO + S \rightarrow_C r$ for all $r \in AL^+$. Moreover, we say that S is a minimal complete set over C , denoted by mcs (resp., maximally incomplete set over C , denoted by MIS) if and only if it is complete (resp., incomplete) over C , and, every proper subset (resp., every strict superset) of S is incomplete (resp., complete) over the same class.

Table 2. Minimal complete and maximal incomplete sets in the class of all linearly ordered sets

MISs	mcss
$\{s\}$	$\{m\}$
$\{f\}$	$\{o, s\}$
$\{o, d, b\}$	$\{o, f\}$
	$\{d, s\}$
	$\{d, f\}$
	$\{s, f\}$
	$\{s, b\}$
	$\{f, b\}$

3 The Class Lin of All Linear Orders

In this section we provide a complete classification of all subsets of AL^+ in terms of expressive completeness over the class of all linear orders (Lin). As it turns out, there are exactly eight mcss, and exactly three MISs, all of them shown in Tab. 2. The rest of this section is devoted to prove that Tab. 2 is correct.

Theorem 1. *In the strict semantics, over the class Lin the MISs and mcss are all and only those shown in Tab. 2.*

In order to prove Theorem 1, we will proceed as follows. First of all, we notice that $\{m\}$ is complete (and clearly, it is minimal), by simply recalling Allen and Hayes’ result [AH85]. Then, we will sketch the completeness of all sets in the right-hand column of the table. Next, we will show that the three sets $\{s\}$, $\{f\}$, and $\{o, d, b\}$ are incomplete; notice that all subsets of an incomplete set are incomplete. Finally, we observe as each proper subset S' of any set S from the right-hand column is incomplete by the previous result, and that each superset S'' of any set of the left-hand column either is complete or it contains a complete set, and we are done. Also, notice that completeness results on a class C of semantic structures are also completeness result on every subclass C' , and every incompleteness result on a class C' also applies to every superclass C .

Lemma 1. *Each set S in the rightmost column of Tab. 2 is complete over the class Lin.*

Proof. The following relation will be useful in the rest of the proof: $r = m \vee b$. Notice that $FO + \{r\} \rightarrow m$, since we have $m(x, y) \leftrightarrow r(x, y) \wedge \neg \exists k(r(x, k) \wedge r(k, y))$. Now, the case $S = \{m\}$ has been proved in [AH85] for the class of all unbounded dense linear orders; it is easy to check that no essential use of density or of the unboundedness is made. As for the case $S = \{s, f\}$, consider the following definability equation:

$$r(x, y) \leftrightarrow \exists k(s(x, k) \wedge f(y, k)) \wedge \neg \exists k(f(k, x) \wedge s(k, y)).$$

We denote the right-hand part of the formula by $\varphi(x, y)$. Assume first that $\mathcal{F} \models \varphi([a, b], [c, d])$. We wish to show that $\mathcal{F} \models r([a, b], [c, d])$, i.e., that $b \leq c$. Suppose, by way of contradiction, that $c < b$. By assumption, there exists an interval $k = [k_1, k_2]$ such that $a = k_1 < b < k_2$ and $k_1 < c < d = k_2$. Then $a < c < b$ and $c < b < d$, hence $[c, b]f[a, b]$ and $[c, b]s[c, d]$, contradicting $\mathcal{F} \models \neg \exists k(f(k, [a, b]) \wedge s(k, [c, d]))$. Conversely, suppose that $\mathcal{F} \models r([a, b], [c, d])$, i.e., $a < b \leq c < d$. Then the interval $k = [a, d]$ witnesses the first conjunct of φ . Moreover, any interval $[a', b]$ finishing $[a, b]$ is disjoint from $[c, d]$, and hence does not start it. The **case** $S = \{o, s\}$ can be dealt with by means of the following equations:

$$r(x, y) \leftrightarrow \neg o(x, y) \wedge \forall z(s(y, z) \rightarrow \neg o(x, z)) \wedge \left(\begin{array}{l} \exists k(s(x, k) \wedge \neg \exists k(s(y, k)) \wedge \neg \exists k(s(k, y))) \\ \vee \exists k_1 \exists k_2(s(x, k_1) \wedge s(y, k_2) \wedge o(k_1, k_2)) \\ \vee \exists k(s(x, k) \wedge o(k, y)) \end{array} \right)$$

The intuition is that we consider three cases, namely (1) y is a unit interval ending with the greatest (end) point in the linear order, (2) y does not end with the greatest point in the linear order, and (3) y is not a unit interval. It should be clear these cases are exhaustive, since the disjunction of (2) and (3) is equivalent to the negation of (1). The top, middle, and lower disjuncts in the last conjunct of the formula will hold, respectively, in cases (1), (2), and (3).

As for the **case** $S = \{o, f\}$ we have that, since oi is definable in terms of o , it becomes precisely symmetric to the case $\{o, s\}$, and we can obtain ri (the inverse relation of r , defined above) in terms of oi and f , which allows us to define mi and hence m . In the **case** $S = \{s, d\}$, we first define o , and then we obtain completeness from the completeness of $\{o, s\}$; to define o , it is sufficient to consider the following definability equation:

$$o(x, y) \leftrightarrow \exists k(s(x, k) \wedge \neg d(y, k) \wedge \exists w(d(w, k) \wedge s(w, y))) \wedge \exists w(s(w, y) \wedge \forall k(d(x, k) \rightarrow d(w, k))).$$

The **case** $S = \{f, d\}$ is symmetric to $\{s, d\}$, and in the **case** $S = \{s, b\}$, we can show that $\{s, b\}$ can define d , and then completeness will follow from the completeness of $\{s, d\}$ which was proven above. This can be done via showing that $\{s, b\}$ can define $r' = d \vee f$. (Note that $r'([a, b], [c, d])$ iff $d([a, b], [c, d]) \vee f([a, b], [c, d])$ iff $c < a < b \leq d$.) It is then immediate to see that the following definition is correct:

$$d(x, y) \leftrightarrow \exists k(s(x, k) \wedge r'(k, y)).$$

It thus remains for us to show how to define r' in terms of s and b , which can be done by means of the following definition:

$$\begin{aligned}
 r'(x, y) \leftrightarrow & \psi(x, y) \wedge \\
 & \exists z(s(z, y)) \wedge \\
 & \forall z((s(z, x) \vee z = x) \rightarrow \neg s(z, y)) \wedge \\
 & \forall k(b(y, k) \rightarrow b(x, k)) \wedge \\
 & (\neg \exists k(s(y, k)) \vee \exists k(s(x, k)))
 \end{aligned}$$

where

$$\begin{aligned}
 \psi(x, y) := & (\neg \exists k(b(k, y)) \wedge \forall z(\neg \exists k(b(k, z)) \rightarrow z = y \vee s(z, y) \vee s(y, z))) \vee \\
 & (\exists k(b(k, x)) \wedge \forall z(b(z, y) \rightarrow b(z, x))).
 \end{aligned}$$

Finally, the **case** $S = \{f, b\}$ is symmetric to the previous one.

Lemma 2. *Each set S in the leftmost column of Tab. 2 is incomplete for the class Lin .*

Proof. We first prove the incompleteness in the **case** $S = \{s\}$. Consider the structure $\mathcal{F} = \langle \mathbb{I}(\mathbb{Q}), s \rangle$, where \mathbb{Q} is the set of rationals with their usual ordering. Define $\zeta : \mathbb{I}(\mathbb{Q}) \rightarrow \mathbb{I}(\mathbb{Q})$ such that

$$\zeta : [a, b] \mapsto [a, a + 2 \cdot |b - a|].$$

In other words, the image of any interval $[a, b]$ under ζ has the same beginning point, but double the length of $[a, b]$. We claim that ζ is an automorphism of the structure \mathcal{F} . It is clear that ζ is a bijection. Further, $[a_1, b_1]s[a_2, b_2]$ if and only if $a_1 = a_2$ and $b_1 < b_2$, that is, if and only if $a_1 = a_2$ and $a_1 + 2 \cdot |b_1 - a_1| < a_2 + 2 \cdot |b_2 - a_2|$, which happens if and only if $\zeta([a_1, b_1])s\zeta([a_2, b_2])$. Now, we show that $FO + \{s\} \not\vdash_C b$, for which it is enough to observe that, since $\zeta([0, 1]) = [0, 2]$ and $\zeta([2, 3]) = [2, 4]$, for all formulas $\varphi(x, y)$ of $FO + \{s\}$ we have that $\mathcal{F} \models \varphi([0, 1], [2, 3])$ if and only if $\mathcal{F} \models \varphi([0, 2], [2, 4])$, but, at the same time $[0, 1]b[2, 3]$ and $\neg([0, 2]b[2, 4])$. A symmetric construction proves the incompleteness of the **case** $S = \{f\}$. For the **case** $S = \{o, b, d\}$ it suffices to consider the structure $\mathcal{F} = \langle \mathbb{I}(\mathbb{D}), o, b, d \rangle$ where \mathbb{D} is the subset $\{-1, 0, 1\}$ of \mathbb{Z} with the usual ordering. An automorphism of this structure can be defined by taking $\zeta : \mathbb{I}(\mathbb{D}) \rightarrow \mathbb{I}(\mathbb{D})$ such that $\zeta([-1, 1]) = [-1, 1]$, $\zeta([-1, 0]) = [0, 1]$, and $\zeta([0, 1]) = [-1, 0]$.

As we observed at the beginning of the section, Lemma 1 and Lemma 2 suffices to prove Theorem 1.

4 The Class Den of All Dense Linear Orders

We now turn to the class of all dense linear orders. The MIS's and mcs's in this case are listed in Tab. 3.

Theorem 2. *In the strict semantics, over the class Den the MISs and mcss are all and only those shown in Tab. 3.*

Table 3. Minimal complete and maximal incomplete sets over the class of dense linear orders

MISs	mcss
{s}	{m}
{f}	{o}
{d}	{b}
	{d, s}
	{d, f}
	{s, f}

The theorem follows immediately from Lemmas 3 and 4, below.

Lemma 3. *Each set S in the rightmost column of Tab. 3 is complete for the class Den.*

Proof. Every complete set over Lin is also complete over every subclass, and hence specifically over Den. Thus that completeness of {m}, {d, s}, {d, f} and {s, f} over Den follow from Lemma 1. It thus suffices to consider the cases {o} and {b}. As for the case $S = \{o\}$, it is sufficient to consider the following definability equation:

$$m(x, y) \leftrightarrow \neg(o(x, y)) \wedge \forall k(o(x, k) \rightarrow \exists w(o(x, w) \wedge o(w, y) \wedge o(w, k))).$$

As for the case $S = \{b\}$, we first define the relation f; by symmetry, it is then easy to define s. Then using the completeness of {s, f}, the result follows. To define f, it is enough to consider the following formula:

$$f(x, y) \leftrightarrow \forall k(b(x, k) \leftrightarrow b(y, k)) \wedge \exists k(b(k, x) \wedge \neg b(k, y)).$$

Lemma 4. *Each set S in the leftmost column of Tab. 3 is incomplete for the class Den.*

Proof. The incompleteness of {s} and {f} can be proved using the same argument as in Lemma 2, since the structure used there was based on the dense linear order Q. For the case {d}, consider the structure $\mathcal{F} = \langle \mathbb{I}(\mathbb{Q}), d \rangle$. Define an $\zeta : \mathbb{I}(\mathbb{Q}) \rightarrow \mathbb{I}(\mathbb{Q})$ such that $\zeta : [a, b] \mapsto [-b, -a]$. We claim that ζ is an automorphism of \mathcal{F} . Indeed, ζ is clearly a bijection, and further, we have that $[a_1, b_1]d[a_2, b_2]$ if and only if $a_2 < a_1 < b_1 < b_2$, that is, if and only if $-b_2 < -b_1 < -a_1 < -a_2$. This happens if and only if $[-b_1, -a_1]d[-b_2, -a_2]$, or, in other words, if and only if $\zeta([a_1, b_1])d\zeta([a_2, b_2])$. Now, we want to show that $FO + \{d\} \not\vdash_{\text{Den}} b$, and for that it is enough to notice that, since $\zeta([0, 1]) = [-1, 0]$ and $\zeta([2, 3]) = [-3, -2]$, for all formulas $\varphi(x, y)$ of $FO + \{d\}$ (even plus equality) we have that $\mathcal{F} \models \varphi([0, 1], [2, 3])$ if and only if $\mathcal{F} \models \varphi([-1, 0], [-3, -2])$. But $[0, 1]b[2, 3]$ and, at the same time, $[-1, 0]b[-3, -2]$. Therefore, the set {d} is not complete.

Therefore, Theorem 2 is proven as a consequence of Lemma 3 and Lemma 4.

5 The Class Dis of All Discrete Linear Orders

The minimal complete (resp., maximal incomplete) sets over the class of all discrete linear orders are identical to those over the class Lin of all linear orders, as shown in Tab. 2. On the one hand completeness transfers from the more general to the less general case. On the other hand, as far as incompleteness is concerned, the incompleteness proof for the $\{o, d, b\}$ (Lemma 2) was based on a discrete structure, and so applies also over Dis. Thus, in order to prove Theorem 3, it thus only remains to show the incompleteness of $\{s\}$ and $\{f\}$. This is done in Lemma 5 below.

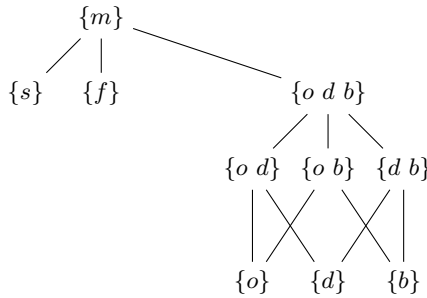


Fig. 1. Expressively different fragments of $FO + AL^+$ in the class of all linear orders

Theorem 3. *In the strict semantics, over the class Dis the MISs and mcss are all and only those shown in Tab. 2.*

Lemma 5. *The sets $\{s\}$ and $\{f\}$ are incomplete over Dis.*

Proof. We only show that $\{s\}$ is incomplete over Dis, the case for $\{f\}$ being treatable in a completely symmetric way. Consider the structure $\mathcal{F} = \langle \mathbb{I}(\mathbb{Z}), s \rangle$ where \mathbb{Z} is the integers with their usual ordering. Define the map $\xi : \mathbb{I}(\mathbb{Z}) \rightarrow \mathbb{I}(\mathbb{Z})$ such that $\xi([a, b]) = [-a, -a + |b - a|]$. Thus ξ reflects each interval's starting point about 0 and maintains its length. It is clear that ξ is a bijection. Furthermore ξ respects the relation s . Indeed, $[a, b]s[c, d]$ iff $a = c$ and $b < c$ iff $-a = -c$ and $(-a + |b - a|) < (-c + |d - c|)$ iff $\xi([a, b])s\xi([c, d])$. Thus ξ is an automorphism of \mathcal{F} . Now $[0, 1]m[1, 2]$ but $\xi([0, 1]) = [0, 1]$, $\xi([1, 2]) = [-1, 0]$ and $\neg([0, 1]m[-1, 0])$. We conclude that $FO + \{s\} \not\vdash_{\text{Dis}} m$.

6 Relative Expressivity

In the foregoing sections we identified all minimally complete and maximally incomplete fragments of first-order logic with equality enriched with Allen's interval relations. This was done over the three classes of linear orders Lin, Den, and Dis. We still have not completely answered the question of which are the expressively different fragments in each of these cases. We will now do so.

Theorem 4. *Over Den there are 4 expressively different fragments of $FO + AL$, namely $\{s\}$, $\{f\}$, $\{d\}$, and $\{m\}$.*

Proof. This is essentially a corollary of Theorem 2; see also Tab. 3. Firstly, all complete sets are expressively equivalent, by definition, and are thus all represented by $\{m\}$. Secondly, all maximally incomplete singletons must be incomparable. Since all subsets with two or more elements are complete, this covers all cases.

Theorem 5. *Over Lin and Dis there are 10 expressively different fragments of $FO + AL$, namely $\{s\}$, $\{f\}$, $\{o\}$, $\{d\}$, $\{b\}$, $\{o, d\}$, $\{o, b\}$, $\{d, b\}$, $\{o, d, b\}$, and $\{m\}$.*

Proof. By Theorems 1 and 3, the set $\{m\}$ is complete and hence represents all complete fragments. The fragments $\{s\}$, $\{f\}$ and $\{o, d, b\}$ are maximally incomplete and hence pairwise incomparable among each other and also different from $\{m\}$. The fragments $\{o\}$, $\{d\}$, $\{o, d\}$, $\{o, b\}$, and $\{d, b\}$ are related as illustrated in the Hasse diagram in Fig. 1. In other words, among these fragments the relative expressivity is simply given by set containment. To see that no two of them are expressively equivalent, it is sufficient to consider the following cases. (Notice that all structures used below are discrete, so the results apply both to Lin and Dis.). As for the case $FO + \{b, d\} \not\vdash_{Dis} o$, consider the structure $\mathcal{F} = \langle \mathbb{I}(\mathbb{D}), b, d \rangle$, where \mathbb{D} is the linear order $0 < 1 < 2 < 3$, and an automorphism ζ which swaps $[0, 2]$ and $[1, 3]$, and is the identity map on all other intervals. This respects d and b , as the only pair in relation d is $([0, 3], [1, 2])$ and the only pair in the relation b is $([0, 1], [2, 3])$, on all terms of which ζ is the identity. However $o([0, 2], [1, 3])$ but $\neg o(\zeta([0, 2]), \zeta([1, 3]))$. In the case $FO + \{o, b\} \not\vdash_{Dis} d$, again, consider the the structure $\mathcal{F} = \langle \mathbb{I}(\mathbb{D}), b, d \rangle$, where \mathbb{D} is the linear order $0 < 1 < 2 < 3$, and an automorphism ζ which swaps $[0, 3]$ and $[1, 2]$, and is the identity map on all other intervals. This respects o and b , as the only pair in relation o is $([0, 2], [1, 3])$ and the only pair in the relation b is $([0, 1], [2, 3])$, on all terms of which ζ is the identity. However $d([1, 2], [0, 3])$ but $\neg d(\zeta([1, 2]), \zeta([0, 3]))$. In case $FO + \{o, d\} \not\vdash_{Dis} b$, consider the the s tructure $\mathcal{F} = \langle \mathbb{I}(\mathbb{D}), o, d \rangle$, where \mathbb{D} is the linear order $0 < 1 < 2 < 3$, and the an automorphism ζ which swaps $[0, 1]$ and $[2, 3]$, and is the identity map on all other intervals. Note that $[0, 1]$ and $[2, 3]$ do not stand in the relations o or d with each other or any other intervals. Thus ζ respects the relations o and d , but violates b as $b([0, 1], [2, 3])$ but $\neg b(\zeta([0, 1]), \zeta([2, 3]))$.

7 Conclusions and Open Problems

In this paper, we have considered extensions of first-order logic with equality with subsets of Allan’s interval relations. We obtained a complete classification of these fragments in terms of relative expressivity when interpreted over the classes of interval structures based on, respectively, all, all dense, and all discrete linear orders. Our results are specific to the strict semantics. Similar results for

the non-strict semantics can be obtained, but, in that case, a closer analysis of the relations is needed, since mixing points and intervals at the algebraic level raises the problem of obtaining a set of mutually exclusive relations. As a consequence of this work, we now have a complete view of the open representation problems in the strict semantics. Another natural question to ask is what happens when equality is not assumed, but treated on the same footing as the other Allen's relations. Finally, one might ask what happens in terms of expressive power when the first order language is limited, say, in the number of variables at disposal, or to some well-defined prefix-quantifier fragment.

Acknowledgements. The work of the first author was supported by grant number 70554 and 70427 of the National Research Foundation of South Africa. The work of the second author has been partially supported by the Spanish project *TIN2009-14372-C03-01*.

References

- [AH85] Allen, J.F., Hayes, P.J.: A common-sense theory of time. In: Proceedings of the 9th International Joint Conference on Artificial Intelligence, pp. 528–531. Morgan Kaufmann (1985)
- [All83] Allen, J.F.: Maintaining knowledge about temporal intervals. *Communications of the ACM* 26, 832–843 (1983)
- [Coe09] Coetzee, C.J.: Representation theorems for classes of interval structures. Master's thesis, Department of Mathematics, University of Johannesburg (2009)
- [GMG03] Goranko, V., Montanari, A., Sciavicco G. Propositional interval neighborhood temporal logics. *Journal of Universal computer science* 9(9), 1137–1167 (2003)
- [GMG04] Goranko, V., Montanari, A., Sciavicco, G.: A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics* 14(1-2), 9–54 (2004)
- [Lad78] Ladkin, P.: The Logic of Time Representation. PhD thesis, University of California, Berkeley (1978)
- [vB83] van Benthem, J.F.A.K.: The logic of time. *Synthese Library*, vol. 156. D. Reidel Publishing Co., Dordrecht (1983); A model-theoretic investigation into the varieties of temporal ontology and temporal discourse
- [Ven90] Venema, Y.: Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic* 31(4), 529–547 (1990)

Using the Relaxed Plan Heuristic to Select Goals in Oversubscription Planning Problems*

Angel García-Olaya, Tomás de la Rosa, and Daniel Borrajo

Universidad Carlos III de Madrid,
Avda de la Universidad 30, 28911, Leganés, Spain

www.plg.inf.uc3m.es

Abstract. Oversubscription planning (OSP) appears in many real problems where finding a plan achieving all goals is infeasible. The objective is to find a feasible plan reaching a goal subset while maximizing some measure of utility. In this paper, we present a new technique to select goals “a priori” for problems in which a cost bound prevents all the goals from being achieved. It uses estimations of *distances* between goals, which are computed using relaxed plans. Using these distances, a search in the space of subsets of goals is performed, yielding a new set of goals to plan for. A revised planning problem can be created and solved, taking into account only the selected goals. We present experiments in six different domains with good results.

1 Introduction

In classical planning the objective is to find a sequence of actions transforming a given initial state into a final state in which a conjunctive list of goals is present. A valid plan is the sequence of actions reaching all goals. Soft goals can be added to a classical planning problem to account for goals that we wish to achieve but that we do not enforce. As a result, a planning problem could contain both hard and soft goals. In real domains there can be several causes making impossible or useless to reach all soft goals: two or more soft goals could be mutually exclusive, i.e. cannot be true at the same time; plans achieving all goals could need more quantity of a certain resource than the available one; goals could be redundant, so a plan would be valid even achieving just some of them; or some goals could be not worth enough, as the cost of achieving them would be higher than their reward. In problems with only soft-goals, a valid plan is any plan achieving any subset of them, even an empty one. Usually, an utility or penalization is assigned to each soft goal to compare plans achieving different sets of soft goals.

Oversubscription planning (OSP) is a special type of planning with soft goals, introduced by [13], and motivated by some real problems at NASA. OSP assumes it is not possible to achieve all soft goals due to a resource limitation; the rover battery power in the original formulation. A simple way to model the limited

* This work has been partially supported by MICIIN TIN2008-06701-C03-03 and CCG10-UC3M/TIC-5597 projects.

resource is as the maximum cost a valid plan can have. The objective is to find a plan that maximizes the utility while keeping the cost (resource) under a certain bound. We will call it the COST-BOUNDED OSP problem to distinguish it from the more general OSP case, where there can be other causes preventing all the goals from being achieved (mutually exclusive goals for example). The objective of COST-BOUNDED OSP problems is to find the plan with maximum *utility* given the resource(s) availabilities. The *utility* is a function, generally the addition, of the utilities of the goals reached by a plan.

A close related soft goals problem is the Partial Satisfaction one (PSP) [12,11,2] included in the International Planning Competition (IPC) 2006 under the PREFERENCES Track and in the IPC 2008 under the NET-BENEFIT track. In PSP, nothing prevents, at least a priori, achieving all goals, but there is a trade-off between the utility of achieving a goal and the cost of doing so [1]. The most explored PSP problem is the NET-BENEFIT, which tries to maximize the *utility* – *cost* metric (actually it assigns a penalization to each not reached goal and minimizes the *cost* + *penalization*). Probably due to the IPC, work on NET-BENEFIT has been extensive. In contrast, the COST-BOUNDED problem has been less explored, even though the existence of a limited resource (time, fuel, battery, storage space, money...) preventing the accomplishment of all goals is present in a large number of real domains.

The optimal solution for a soft goals problem can be computed by finding a plan for each of the 2^n combinations of the n problem's goals, and then selecting the plan with maximum utility. Of course, this is infeasible except for very simple cases. In practice, three different approaches have been used: a priori selection of goals, on-the-fly selection, and compilation into a different problem. Goals can be selected "a priori" to find the potentially best subset to plan for. Later, the planning step takes into account only the selected goals. In [13], an orienteering problem (OP) is constructed. A set of propositions, different for each domain, make up the nodes of the basic OP. For each goal a node is added, inserting an arc from it to the nodes where it can be achieved. Arcs costs are calculated using a plan graph. The resulting OP is solved using beam search and the solution is used to guide a partial-order planner. This is the only approach tailored for resource-bounded problems found in the literature. The main disadvantage is that the set of propositions making up a node depends on a threshold that has to be manually defined for each domain. In [12], relaxed plans are used to estimate the cost of reaching a goal, but for the NET-BENEFIT problem. For a n – *goals* problem, a relaxed plan to each goal is computed to estimate the NET-BENEFIT of including it in the set of goals. Then, for each goal, a set containing it is constructed adding goals until the NET-BENEFIT does not increase. On-the-fly selection of goals approaches do not perform goals selection. Instead, incremental plans are built, refining the best ones to achieve more utility [4,1]. In general, these approaches scale worse than the goal selection ones. A soft goals problem can be modeled as a Markov Decision Process (MDP) [13,2], obtaining a policy from which a plan finding the optimal solution can be extracted. However, this conversion does not scale well so it has not been reported to be used

in practice. Using integer programming (IP) to find optimal plans for a given parallel plan length is another possible transformation [2]. The most successful problem transformation [9] compiles away soft goals to create a STRIPS+actions cost problem with more actions, fluents and hard goals, but no soft goals, which can be solved by any conventional planner. This compilation, in combination with the winner of the satisficing track of the IPC 2008 [11], outperforms any participant of the soft-goals tracks of the two last IPCs [9].

The technique we propose is a two-step algorithm. In the first step we select the goals to plan for by using relaxed plans to compute distances among goals. Distances represent estimated costs of achieving one goal from a state where another one has been achieved. Using those distances, we perform a search in the space of subsets of goals for the set that maximizes the utility with the estimated cost-bound. In the second step, this set is given to a satisficing planner to find a plan achieving it.

In the following section we will formally define the problem. Next, the two-step algorithm will be described, and our technique will be compared with previous work. The paper finishes with conclusions and future work.

2 Problem Definition

We will define next the planning problem we are tackling.

Definition 1: A STRIPS planning problem with actions costs and soft goals is a tuple $P = \{F, A, I, G, c, u\}$, where F is a finite set of fluents, A is a finite set of actions, being each $a_i \in A$ composed of preconditions establishing when the action can be applied, and effects, consisting of elements of F being added or deleted from the current state after a_i is applied, $I \subseteq F$ is the initial state, $G \subseteq F$ is the set of goals, $c : A \mapsto \mathbb{R}_0^+$ is a cost function, and $u : G \mapsto \mathbb{R}^+$ is an utility function.

A solution of the planning problem P is an ordered list of actions $\Pi = \{a_0, a_1 \dots a_n\}$, $a_i \in A$, which applied to I results in a state where $G' \subseteq G$ is true (in classical planning, $G' = G$). If the final state is forced to achieve some $G'' \subset G'$ then we have a problem with both hard and soft goals. The cost of the plan Π is defined as $C(\Pi) = \sum_{a_i \in \Pi} c(a_i)$. The objective of a planning problem with soft goals is usually to maximize the utility of the plan. We will consider additive utilities, as most work in the field (see [4] for other approximations).

Definition 2: The utility of a solution, Π , to the planning problem with soft goals is $U(\Pi) = \sum_{g_i \in G'} u(g_i)$.

Definition 3: A COST-BOUNDED problem is a tuple $M = \{P, C_{max}\}$, where P is a planning problem with soft goals as defined above, and $C_{max} \in \mathbb{R}^+$ is the cost bound of the problem.

A solution to the COST-BOUNDED problem is a plan $\Pi = \{a_0, a_1 \dots a_n\}$, $a_i \in A$, which applied to I results in a state where $G' \subseteq G$ is true, and such that its plan cost satisfies $C(\Pi) \leq C_{max}$. Given two solutions for the COST-BOUNDED problem Π_1 and Π_2 , Π_1 will be a better solution than Π_2 if $U(\Pi_1) > U(\Pi_2)$.

3 Two-Step OSP Algorithm

We perform an *a priori* selection of goals in two steps: selecting goals and planning for the selected goals. Algorithm 1 shows the pseudo-code of the process.

Algorithm 1. Two-step algorithm for solving COST-BOUNDED problems

```

OSP (P OSP problem):  $\Pi$  Plan
   $S \leftarrow \text{Select-goals}(P)$ 
   $P' \leftarrow$  standard problem (no OSP) from new goals( $P, S$ )
   $\Pi \leftarrow \text{Plan-for-goals}(P')$ 
return  $\Pi$ 

Select-goals (P OSP problem):  $S$  Goals set
   $D \leftarrow$  Compute distances matrix
   $S \leftarrow \text{Select goals}(D)$ 
return  $S$ 

Plan-for-goals (P planning problem):  $\Pi$  plan
repeat
   $\Pi \leftarrow \text{plan}(P)$ 
  if  $\Pi \neq \emptyset$  then
    return  $\Pi$ 
  else
     $P \leftarrow$  remove lowest utility goal from  $P$ 
  end if
until  $\text{goals}(P) = \emptyset$ 
return Fail

```

3.1 Selecting Goals

In the first step, we generate a matrix of distances between goals. This matrix has $n + 1$ rows and n columns, being n the number of goals. The elements of the first row are the estimations of the cost of reaching each goal from I , as if each goal were the only goal in the problem. The following rows, one for each goal, contain the estimations of the cost of achieving the remaining goals from the state reached when calculating the first row.

Definition 4 (Distance from the initial state to a goal): *Let P be a planning problem and $g_x \in G$ a goal. We define the distance from I to g_x (Δ_{Ix}) as the cost of the lowest cost plan, Π_x^* reaching g_x from I .*

The value of Δ_{Ix} gives an idea about how close a given goal and the initial state are, i.e. how costly to reach a single goal from the initial state is. Therefore, it makes it easier to decide whether to include or not this goal in the set of goals to plan for. In most cases, achieving one goal will change the cost of reaching others, which is accounted for by means of the following distance between two goals:

Definition 5 (Distance between two goals): Let P be a planning problem, $g_x \in G$ a goal, Π_x^* the lowest cost plan used to compute $\Delta_{I,x}$, $s_{\Pi_x^*}$ the state resulting from applying Π_x^* to I , and $g_y \in G$ another goal. The distance from g_x to g_y ($\Delta_{x,y}$) is defined as the cost of the lowest cost plan reaching g_y from $s_{\Pi_x^*}$.

So, in order to compute the distance between two goals, the lowest cost plan computed in the previous step is applied to the initial state to reach another state where the first goal is achieved, and then a distance to the second goal is computed in the same way as before. In general, $\Delta_{x,y} \neq \Delta_{y,x}$.

Both $\Delta_{I,x}$ and $\Delta_{x,y}$ depend on the calculation of the lowest cost plan with only one goal. Even if all the other goals are removed, computing this plan is usually difficult, making the computation of these distances infeasible and discouraging their use. As an example, we tried to use an optimal planner to compute $\Delta_{I,x}$ for the propositional Rovers domain. It was only possible to compute it in the first 22 (out of 40) IPC5 problems.

Instead, an approximation of $\Delta_{I,x}$ ($\Delta'_{I,x}$) can be computed using relaxed plans. For the rest of the paper, we will compute $\Delta'_{I,x}$ as the cost of the non-optimal relaxed plan reaching g_x from I , in a similar way as Metric-FF does [8]. In order to compute $\Delta'_{x,y}$ the relaxed plan extracted to compute $\Delta'_{I,x}$ is applied to I to reach a state in which g_x is true. Obviously, quite often, actions belonging to the relaxed plan will not be applicable, as some of their preconditions will not be satisfied. Despite this fact, actions are applied ignoring preconditions and only taking into account the effects. From this state, the distance to g_y is computed, using again the cost of the non-optimal relaxed plan. A mutex check should be done before calculating $\Delta'_{x,y}$, but as there are no mutex goals in the domains we have tried, we have not implemented it yet.

Once the distances matrix is generated, we use a beam search algorithm to find the goal set with higher utility in the space of subsets of goals. The root node is the empty set. In the first step, the k goals with higher utility are selected, being k the beam width. For each of the selected goals g_x , we annotate the node with the corresponding $\Delta'_{I,x}$ and $u(g_x)$. In the second step we consider all the combinations of the previously selected k goals and one of the remaining goals and annotate the accumulated cost and utility for these two-goals sets. If a set is composed of g_1 and g_2 , the utility will be $u(g_1) + u(g_2)$ and the cost $\Delta'_{I,g_1} + \Delta'_{I,g_2}$. We select the k best sets and so on. Search ends when a set including all the goals has been found, or, more likely, when it is not possible to add goals to any of the k best sets without exceeding the cost-bound. In this case, the set with higher utility is returned as the solution of the search process. We break ties favoring lower estimated cost. In case of further tie, one is picked arbitrarily. This algorithm is greedy in the sense that once a goal is selected for inclusion in a planning set, it is always considered in the same relative order with respect to the other goals in that set.

Regarding search parameters, we have tried with different beam widths (0.25, 0.5, 1, 5, 10, 50, 100 and 500 times the number of goals) and 5 gives the best results in most domains, although variations in utilities depending on beam width do not seem to be very high.

3.2 Planning for the Selected Goals

Once goals have been selected, we generate a new problem with the goals in the selected goals set. The new problem is given to a Metric-FF-like planner, CBP [6]. Its performance is comparable to LAMA for many domains and, unlike it, it allows numeric preconditions, which are present in all COST-BOUNDED domains.

Given that *distances* (i.e. costs) are estimated, often the list given to the planner is still oversubscribed. To solve this problem, we try to find a plan during a given time bound. If no plan is found, the lowest utility goal is removed and we search for a new plan. This is repeated until a valid plan has been found or all the goals have been removed. To find a plan, the planner is given the same time used to calculate distances, with a minimum of 10 seconds, which has shown experimentally to work well.

3.3 Computational Complexity

For a n -goals problem, we have to extract n non optimal relaxed plans to compute Δ'_{I_x} , which is polynomial in time [7]. Then we apply these relaxed plans to obtain the new initial states for each goal, which is linear in the number of steps of the relaxed plans. Next, for each goal g_x , we have to create $n-1$ relaxed plans to compute Δ'_{x_y} . This gives us $n + n \times (n - 1)$ relaxed plans, so it is quadratic in the number of relaxed plans. Comparing with [12], in the worst case, when low oversubscription, the complexity of their approach in terms of relaxed plans and goals is $n \times \sum_{i=1}^{n-1} i \approx n^3$ relaxed plans. Furthermore, we are always solving relaxed plans with only one goal while they incrementally construct relaxed plans with two, three, ... goals. If only a few of the goals can be achieved, the complexity of their algorithm decreases dramatically, while ours remains constant. In addition, we have to select the goals. Nodes at the first layer of the search graph include only one goal. In the second, they have two goals, and so on. Thus, the maximum depth will be $n-1$ in case all the goals except one can be achieved. Given that we have chosen a beam search width of $5n$, the complexity is quadratic in n . In comparison, complexity of [13] OP is exponential on the number of propositions defining a node, which depends on a manually selected threshold and varies in each domain.

4 Experimental Results

We have tested our technique in six IPC domains. To create COST-BOUNDED problems, first we have taken the PREFERENCES or NET-BENEFIT versions, removed the preferences part to make them regular actions cost domains and solved them using a Metric-FF like satisficing planner. The aim is to have an upper bound for the plan cost. In domains where no such versions exist we have used the STRIPS + actions cost version, solving it in the same way. Second, equivalent problems with a cost bound of 25%, 50% and 75% of the previously computed cost have been generated. These three values allow to study how well

our technique performs when there is a high (25%), medium (50%) or low (75%) oversubscription degree. Domains have been changed by adding a new fluent to account for the cost bound. For any action increasing the cost of the plan a new precondition has been added. This precondition prevents the action from being applied if its cost, plus the current accumulated cost, exceeds the cost bound. For example, if the cost of a problem solution is 100, three new problems have been created. These problems have the same initial state and goals than the original one, but their maximum cost is limited to 25, 50 and 75 respectively. We did not use the penalizations of the original problems because we were interested in testing whether different distributions of utilities among goals yield different results. Instead, we have defined two versions of each problem; in the first one all the goals have the same utility: $u(g_i) = 1, \forall g_i \in G$. In the second one, the utility of each goal is a random value between 1 and 10: $1 \leq u(g_i) \leq 10, \forall g_i \in G$.

We have compared our approach, that we will call *Distances*, with the most similar current work: an adaptation for COST-BOUNDED problems of Keyder et al. compilation [9]; Mips-XXL [5], ranked second in the NET-BENEFIT track of the IPC 2008 (the winner exhausts the memory even with the simpler problems); SGPlan [3], winner of the PREFERENCES track of the IPC 2006; and a *Baseline* planner which greedily selects the goal with higher utility and plans for it. If a plan is found, the two goals with higher utility are selected and so on. Compiled, Mips-XXL and SGPlan are tailored for the NET-BENEFIT problem and not for the COST-BOUNDED one. That means that in the search process they will try to minimize the penalization for not achieving the soft goals, but in general the heuristic will go *blind* with respect to the cost bound. A way to tackle this is to modify the metric, so the problem is converted into a kind of NET-BENEFIT-COST-BOUNDED one, i.e. both the total cost and the penalizations have to be minimized. But a focus has to be put on the penalizations as against the cost; the planner should not avoid reaching a goal even if its utility is lower than its cost given that the cost is not higher than the cost bound. As a preliminary version, we have changed the metric of the compiled problems from *(minimize (+ (penalizations-cost))* to *(minimize (+ (penalizations-cost)) / (plan-cost) (cost-bound))*, which slightly improves their performance

Domains tested are *Rovers* from IPC5, and *Driverlog* and *Depots* from IPC3. *Rovers* is a good example of a domain where goals can not be undone once achieved and there are not strong interactions among goals. *Depots* has been chosen because goals can be undone and there are many interactions among them. In *Driverlog*, in addition, a significant percentage of the goals are present at the initial state and the planner must undo them to find a valid plan. We have also tested *Transport*, *Peg Solitaire* and *Elevators* domains from the IPC 2008 NET-BENEFIT track. *Crewplanning* has universal quantifiers not supported by our planner. *Openstacks* is mainly an optimization domain in which the cost of a good plan is very low, making it difficult to create different degrees of oversubscription. And *Woodworking's* soft goals are not a single predicate but a conjunction of them, which is not supported yet by our approximation. For the experiments we have used a Intel Xeon 3Ghz with 3GB of RAM memory and

a time bound of 900 seconds. For the *Compiled* problems, the planner uses the whole 900 seconds to find and refine the plan. The same applies for Mips-XXL, which, although being an optimal planner, is able to generate intermediate non-optimal plans. In contrast, for *Baseline*, SGPlan and the *Distances* version, no plan refining is done; the first found plan is returned.

Table 1 shows the results. Scores for each planner are calculated in a similar manner as in the IPC: the planner finding the plan with higher utility (U_{max}) gets 1 point. Every other planner scores U/U_{max} . SGPlan has been removed from the table as it only solves problems in two domains (*Peg Solitaire* and *Rovers*) and even in these domains the quality is quite poor. The best result for each domain and oversubscription degree is highlighted in bold.

Table 1. Results on quality. Number next to each domain is the number of problems, i.e. the maximum score a planner can get. High (25%), medium (50%) and low (75%) oversubscription rates have been considered.

Domain	<i>Baseline</i>			<i>Distances</i>			<i>Compiled</i>			<i>Mips-XXL</i>		
	25%	50%	75%	25%	50%	75%	25%	50%	75%	25%	50%	75%
<i>Depots</i> ₁ (22)	12.9	14.0	14.8	16.6	16.0	16.4	14.8	14.2	15.8	14.9	12.5	11.4
<i>Depots</i> ₁₀ (22)	12.7	14.8	14.5	15.9	17.6	16.7	13.1	13.1	14.9	14.9	13.0	12.7
<i>Driverlog</i> ₁ (20)	11.1	12.0	13.3	16.4	17.8	18.1	15.9	15.4	14.8	13.5	14.1	12.3
<i>Driverlog</i> ₁₀ (20)	11.8	12.6	14.3	17.0	18.0	18.9	16.1	15.7	14.7	14.0	14.7	13.1
<i>Elevators</i> ₁ (30)	8.0	22.0	22.6	23.5	25.8	28.1	26.0	29.2	28.2	1.5	0.8	0.8
<i>Elevators</i> ₁₀ (30)	11.4	19.9	26.5	22.8	25.4	27.4	26.0	28.7	27.6	23.7	19.2	13.5
<i>Pegsol</i> ₁ (30)	20.2	19.9	20.4	24.9	27.8	29.0	28.2	29.6	29.8	28.2	28.8	27.0
<i>Pegsol</i> ₁₀ (30)	21.0	21.2	22.2	27.3	28.8	29.1	29.0	29.7	29.9	29.0	28.6	27.6
<i>Transport</i> ₁ (30)	11.0	14.8	19.9	15.5	18.8	21.7	13.0	17.3	18.1	0.0	0.0	0.0
<i>Transport</i> ₁₀ (30)	7.7	16.5	23.4	15.0	21.4	24.6	12.6	18.4	19.9	0.0	0.0	0.0
<i>Rovers</i> ₁ (20)	10.5	15.9	17.6	15.2	16.9	19.2	20.0	19.7	19.0	16.3	11.6	9.0
<i>Rovers</i> ₁₀ (20)	10.6	15.9	18.3	14.1	16.9	19.2	20.0	19.7	18.3	17.8	13.8	11.3
Total	148.8	199.4	227.7	222.7	250.8	268.6	234.6	250.8	250.9	173.6	157.0	138.6

Baseline performs always worse than *Distances*, except in some domains, especially with low oversubscription, where it performs closer. In general, IPC soft goals domains tend to have a low number of goals, most of the times less than ten. In this case, the greedy approach of *Baseline* performs close to other approaches when the cost bound is high. In domains that have problems with a higher number of goals, like *Driverlog* or *PegSolitaire*, the differences are much bigger. We plan to create more complicated problems to see if this tendency continues. *Distances* performs better than *Compiled* in 20 problem sets and worse in 16, while Mips-XXL is better only in low oversubscription *PegSolitaire*. Different utility profiles make no significant difference, but degree of oversubscription does. *Compiled* performs better in low oversubscription domains in 8 out of 12 configurations. In high and medium oversubscription degrees there is a tie; both approaches behave better in 6 out of 12. In these problems, the low *cost-bound* prunes quite quickly the search tree, allowing a more complete exploration by

the iteratively refining algorithm. As soon as the maximum cost increases, yielding a bigger search space, selection of goals by *Distances* returns better results. Again, we expect that in more complicated problems these differences will be magnified and *Distances* will outperform *Compiled*.

Time cannot be easily compared as *Compiled* and *Mips-XXL* use the whole 900 seconds to refine the solutions, while the other planners finish as soon as a valid plan has been found. Table 2 shows the accumulated time needed to find the best solution for $util = 1$, problems (results for $1 \leq util \leq 10$ problems are similar). For *Compiled* it means time spent to find the last solution within the 900 seconds limit (so, not necessarily consuming all the time). *Mips-XXL* is not included in the comparison as there is no way to know when the partial solutions are generated. *Baseline* is usually the fastest one, though in some domains *Distances* is better. *Compiled* is most of the times the slowest one.

Table 2. Accumulated total time in seconds to find the best solution

Domain	<i>Baseline</i>			<i>Distances</i>			<i>Compiled</i>		
	25%	50%	75%	25%	50%	75%	25%	50%	75%
<i>Depots</i>	189	255	265	1391	1826	1788	4306	3851	5051
<i>Driverlog</i>	139	203	214	585	879	799	436	1951	3593
<i>Elevators</i>	47	253	321	19	122	185	22	3005	4638
<i>PegSolitaire</i>	10	80	122	910	1525	1766	224	3266	3401
<i>Rovers</i>	189	207	209	59	145	86	141	2617	2779
<i>Transport</i>	131	212	274	195	345	354	1145	3513	2655

5 Conclusions and Future Work

In this paper we have presented a method to solve COST-BOUNDED oversubscription problems based on the computation of a *distance* between goals using relaxed plans. This distance indicates how far two goals are, allowing to search in the space of subsets of goals to find a subset maximizing utility with an estimated cost lower than a given cost bound. To find plans for this subset we have used a planner with performance comparable to the winner of the last IPC.

We have evaluated this approach against NET-BENEFIT planners as no other COST-BOUNDED planner is, to our knowledge, freely available. Problems with high, medium and low oversubscription have been created by limiting the cost a plan can have to 25%, 50% and 75% of the estimated total cost. Results show that our technique offers better quality in problems with low oversubscription and in domains where the medium number of goals is above ten. In problems with high or medium oversubscription or with low number of goals, its performance is comparable with the best technique; Keyder’s et al. compilation. Our technique is also almost always much faster than the compilation.

In the future, we plan to implement smarter strategies to apply when the selected goals set is still oversubscribed, or when the real cost of the found plan is lower than the maximum cost, allowing thus for more goals to be achieved.

In our current implementation the planner does not take any advantage of the order in which the goals were selected. We want to explore whether biasing the planner to follow this order would increase the performance. Some ways to do that are, for example, to modify the heuristic values of nodes, or to use a goal agenda as presented in [10].

We plan also to make experiments in other domains and in more complicated problems, as those of the sequential satisficing track of the IPC. In the current configuration, for NET-BENEFIT planners (*Compiled*, *SGPlan* and *Mips-XXL*), the effect of the metric is to take into account the cost of the plan as another goal (for util=1 problems) or as the lowest utility goal (for $1 \leq \text{util} \leq 10$ problems). We want to experiment with different metrics to guide the NET-BENEFIT planners in a different way.

References

1. Benton, J., Do, M., Kambhampati, S.: Anytime heuristic search for partial satisfaction planning. *Artificial Intelligence* 173, 562–592 (2009)
2. van den Briel, M., Sanchez, R., Do, M.B., Kambhampati, S.: Effective approaches for partial satisfaction (over-subscription) planning. In: *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004)*, pp. 562–569 (July 2004)
3. Chen, Y.X., Wah, B.W., Hsu, C.W.: Temporal planning using subgoal partitioning and resolution in sgplan. *J. of Artificial Intelligence Research* 26, 323–369 (2006)
4. Do, M.B., Benton, J., van den Briel, M., Kambhampati, S.: Planning with goal utility dependencies. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, Hyderabad, India, pp. 1872–1878 (2007)
5. Edelkamp, S., Jabbar, S.: Mips-xxl: Featuring external shortest path search for sequential optimal plans and external branch-and-bound for optimal net benet. In: *Proc. 2008 International Planning Competition*, Sydney, Australia (2008)
6. Fuentetaja, R., Borrajo, D., Linares, C.: A look-ahead B & B search for cost-based planning. In: *Proceedings of the Thirteenth Conference of the Spanish Association for Artificial Intelligence*, pp. 105–114 (2009)
7. Hoffman, J., Nebel, B.: The ff planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 14, 253–302 (2001)
8. Hoffmann, J.: The Metric-FF planning system: Translating “ignoring delete lists” to numeric state variables. *Journal of Artificial Intelligence Research* 20, 291–341 (2003)
9. Keyder, E., Geffner, H.: Soft goals can be compiled away. *Journal of Artificial Intelligence Research* 36, 547–556 (2009)
10. Koehler, J., Hoffmann, J.: On reasonable and forced goal orderings and their use in an agenda-driven planning algorithm. *Journal of Artificial Intelligence Research* 12, 338–386 (2000)
11. Ritcher, S., Westphal, M.: The lama planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research* 39, 127–177 (2010)
12. Sanchez-Nigenda, R., Kambhampati, S.: Planning graph heuristics for selecting objectives in over-subscription planning problems. In: *Proceedings of the 15th Intl. Conf. on Automated Planning & Scheduling (ICAPS 2005)*, pp. 192–201 (2005)
13. Smith, D.: Choosing objectives in over-subscription planning. In: *Proceedings of the 14th Intl. Conf. on Automated Planning & Scheduling*, pp. 393–401 (2004)

Optimally Scheduling a Job-Shop with Operators and Total Flow Time Minimization

María R. Sierra, Carlos Mencía, and Ramiro Varela

Department of Computer Science,
University of Oviedo, (Spain) Campus de Viesques s/n, Gijón, 33271, Spain

<http://www.di.uniovi.es/tc>

Abstract. We face the job-shop problem with operators and total flow time minimization. This problem extends the classical job-shop problem by considering a limited number of operators that assist the processing of the operations. We propose a schedule generation scheme that extends the well-known *G&T* algorithm. This scheme is then exploited to design an any-time algorithm that combines best-first and greedy search and takes profit from two monotonic heuristics and a method for pruning states based on dominance relations. The results of an experimental study across several benchmarks show that our approach outperforms a constraint programming approach.

1 Introduction

We face the job-shop scheduling problem with operators and total flow time minimization by means of best-first heuristic search. This problem has been recently proposed in [1] for makespan minimization and is formalized as a classical job-shop problem in which the processing of an operation on a given machine requires the assistance of one of the p available operators. So, this extension brings the job-shop scheduling problem closer to real-world problems. Besides, minimizing the total flow time is very interesting within manufacturing and services sectors, and makes the problem harder to solve [3].

The main contribution of the present paper is the definition and study of a new schedule generation scheme that is inspired in the *G&T* algorithm proposed in [2] for the classical job-shop scheduling problem. This new scheme is then exploited to devise a best-first search algorithm and also a greedy algorithm. These algorithms are used in combination and their performance relies on two heuristic estimations and a rule for pruning search states that are also proposed in this paper. We have conducted an experimental study across instances of different sizes and characteristics. The results of this study show that our approach outperforms a constraint programming approach.

The remaining of the paper is organized as follows. Firstly, we define the problem and propose a disjunctive model for it. Then we present the new schedule generation scheme termed *OG&T*. After that, the new best-first search and greedy algorithms are described. The subsequent section is devoted to the experimental study and finally we provide the main conclusions and propose some ideas for future research.

2 Problem Formulation

Formally the job-shop scheduling problem with operators can be defined as follows. We are given a set of n jobs $\{J_1, \dots, J_n\}$, a set of m resources or machines $\{R_1, \dots, R_m\}$ and a set of p operators $\{O_1, \dots, O_p\}$. Each job J_i consists of a sequence of v_i operations or tasks $(\theta_{i1}, \dots, \theta_{iv_i})$. Each task θ_{il} has a single resource requirement $R_{\theta_{il}}$, an integer duration $p_{\theta_{il}}$ and a start time $st_{\theta_{il}}$ and an assisting operator $O_{\theta_{il}}$ to be determined. A feasible schedule is a complete assignment of starting times and operators to operations that satisfies the following constraints: (i) the operations of each job are sequentially scheduled, (ii) each machine can process at most one operation at any time, (iii) no preemption is allowed and (iv) each operation is assisted by one operator and one operator cannot assist more than one operation at the same time. The objective is finding a feasible schedule that minimizes the sum of the completion times of all the jobs, i.e. the total flow time. This problem was first defined in [1] for makespan minimization and is denoted as $JSO(n, p)$.

The significant cases of this problem are those with $p < \min(n, m)$, otherwise the problem is a standard job-shop problem denoted as $J||\Sigma C_i$.

Scheduling problems are usually represented by means of a disjunctive model. We propose here to use the following model for the $JSO(n, p)$ that is similar to that used in [1]. A problem instance is represented by a directed graph $G = (V, A \cup E \cup I \cup O)$. Each node in the set V represents either an actual operation, or any of the fictitious operations with null processing time. These fictitious operations include starting operations for each operator O_i , denoted O_i^{start} , and the dummy operations *start* and *end*.

The arcs in A are called *conjunctive arcs* and represent precedence constraints among operations of the same job. The arcs in E are called *disjunctive arcs* and represent capacity constraints. E is partitioned into subsets E_i with $E = \cup_{\{i=1, \dots, M\}} E_i$. E_i includes an arc (v, w) for each pair of operations requiring the resource R_i . The set O of *operator arcs* includes one arc (u, v) for each pair of operations of the problem, and arcs (O_i^{start}, u) for each operator node and operation. The set I includes arcs connecting node *start* to each node O_i^{start} . Arcs are weighted with the processing time of the operation at the source node.

From this representation, building a solution can be viewed as a process of fixing disjunctive and operator arcs. A disjunctive arc between operations u and v gets fixed when one of (u, v) or (v, u) is selected and consequently the other one is discarded. An operator arc between u and v is fixed when (u, v) , (v, u) or none of them is selected, and fixing the arc (O_i^{start}, u) means discarding (O_i^{start}, v) for any operation v other than u .

Therefore, a feasible schedule S is represented by an acyclic subgraph of G , of the form $G_S = (V, A \cup H \cup I \cup Q)$, where H expresses the processing order of operations on the machines and Q expresses the sequences of operations that are assisted by each operator. The completion time of job J_i is the length of the longest path from *start* to *end* restricted to pass through node θ_{iv_i} .

Figure 1 shows a solution graph for an instance with 3 jobs, 3 machines and 2 operators. Discontinuous arrows represent operator arcs. So, the sequences

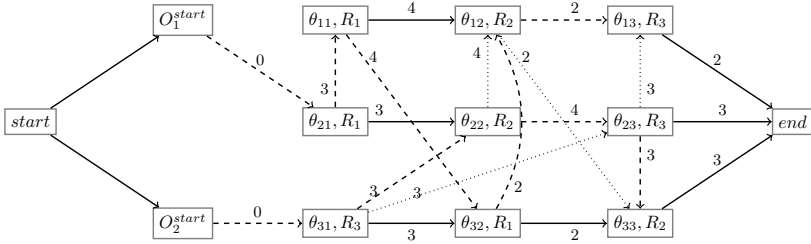


Fig. 1. A feasible schedule to a problem with 3 jobs, 3 machines and 2 operators

of operations assisted by operators O_1 and O_2 are $(\theta_{21}, \theta_{11}, \theta_{32}, \theta_{12}, \theta_{13})$ and $(\theta_{31}, \theta_{22}, \theta_{23}, \theta_{33})$ respectively. In order to simplify the picture, only the operator arc is drawn when there are two arcs between the same pair of nodes. Continuous arrows represent conjunctive arcs and dotted arrows represent disjunctive arcs; in these cases only arcs not overlapping with operator arcs are drawn. In this example, the completion times of jobs J_1, J_2 and J_3 are 13, 10 and 14 respectively, so the schedule has a total flow time of 37.

In order to simplify expressions, we define the following notation for a feasible schedule. The *head* r_v of an operation v is the cost of the longest path from node *start* to node v , i.e. it is the value of st_v . The *tail* q_v is defined so as the value $q_v + p_v$ is the cost of the longest path from v to *end*. PM_v and SM_v denote the predecessor and successor of v respectively on the machine sequence, PJ_v and SJ_v denote the predecessor and successor operations of v respectively on the job sequence and PO_v and SO_v denote the predecessor and successor operations of v respectively on the operator of v .

A partial schedule is given by a subgraph of G where some of the disjunctive and operator arcs are not fixed yet. In such a schedule, heads and tails can be estimated as

$$r_v = \max\left\{ \max_{J \subseteq P(v)} \left\{ \min_{j \in J} r_j + \sum_{j \in J} p_j \right\}, \max_{J \subseteq PO(v)} \left\{ \min_{j \in J} r_j + \sum_{j \in J} p_j \right\}, r_{PJ_v} + p_{PJ_v} \right\} \quad (1)$$

$$q_v = \max\left\{ \max_{J \subseteq S(v)} \left\{ \sum_{j \in J} p_j + \min_{j \in J} q_j \right\}, \max_{J \subseteq SO(v)} \left\{ \sum_{j \in J} p_j + \min_{j \in J} q_j \right\}, p_{SJ_v} + q_{SJ_v} \right\} \quad (2)$$

with $r_{start} = q_{end} = r_{O_i^{start}} = 0$ and where $P(v)$ denotes the disjunctive predecessors of v , so as for all $w \in P(v)$, $R_w = R_v$ and the disjunctive arc (w, v) is already fixed (analogously, $S(v)$ denotes the disjunctive successors of v). $PO(v)$ denotes the operator predecessors of v , i.e $w \in PO(v)$ if it is already established that $O_w = O_v$ and w is processed before v , so as the operator arc (w, v) is fixed (analogously, $SO(v)$ are the operator successors of v).

3 Schedule Generation Schemes

We propose a schedule generation scheme for the $JSO(n, p)$ that is an extension of the well-known $G\&T$ algorithm, proposed by Giffler and Thompson [2] for the classical $J||C_{max}$ problem. The operations are scheduled one at a time in sequential order within each job. When an operation u is scheduled, it is assigned a starting time st_u and an operator $O_i, 1 \leq i \leq p$. Let SC be the set of scheduled operations at an arbitrary time and G_{SC} the partial solution graph built so far. Let A be the set that includes the first unscheduled operation of each job that has at least one unscheduled operation, i.e.

$$A = \{v \notin SC, \nexists PJ_v \vee PJ_v \in SC\} \tag{3}$$

For each operation u in A , r_u is the starting time of u if u is selected to be scheduled next. Let $t_i, 1 \leq i \leq p$, be the time at which the operator O_i is available, then

$$r_u = \max\{r_{PJ_u} + p_{PJ_u}, r_v + p_v, \min_{1 \leq i \leq p} t_i\} \tag{4}$$

where v denotes the last operation scheduled having $R_v = R_u$. In general, a number of operations in A could be scheduled simultaneously at their current heads, however it is clear that not all of them could start processing at these times due to both capacity constraints and operators availability. So, a straightforward schedule generation scheme is obtained if each one of the operations in A is considered as candidate to be scheduled next.

If the selected operation is u , it is scheduled at the time $st_u = r_u$ and any operator O_i with $t_i \leq r_u$ can be selected for the operation u . So, all the operations in A are given the chance to start at their head in A . The next result can be proved that guarantees that the set of schedules generated with this scheme is dominant, i.e. it contains at least one optimal schedule.

Proposition 1. *In at least one of the optimal schedules reachable from G_{SC} there is an operation $u \in A$ that is scheduled at a time $st_u = r_u^A$, where r_u^A is the head of u in G_{SC} .*

Let v^* be the operation in A having the earliest completion time, i.e. $v^* = \arg \min\{r_u + p_u; u \in A\}$. The set of non-deterministic choices may be reduced if we consider the set $A' = \{u \in A; r_u < r_{v^*} + p_{v^*}\}$ instead of A . The reason for this is that for any operation $u \in A \setminus A'$, its head is the same just before and after the operation v^* is scheduled.

We can go further in restricting the choices in each step from the following observations. If the number of operators available is large enough, it is not necessary to take all the operations in the set A' as candidate selections. Let $[T, C)$ be a time interval, where $T = \min\{r_u; u \in A'\}$ and $C = r_{v^*} + p_{v^*}$, and the set of machines $R_{A'} = \{R_u; u \in A'\}$. If we consider the simplified situation where $r_u = T$, for all $u \in A'$ we can do the following reasoning: if, for instance, the

number of machines in $R_{A'}$ is two and there are two or more operators available along $[T, C)$, then the set A' can be reduced to the operations requiring the machine R_{v^*} . In other words, we can do the same as it is done in the $G\&T$ algorithm for the classical job-shop problem. The reason for this is that after selecting an operation v requiring R_{v^*} to be scheduled, every operation $u \in A'$ requiring the other machine can still be scheduled at the same starting time as if it were scheduled before v , so as this machine may not be considered in the current step. However, if there is only one operator available along $[T, C)$ then A' may not be reduced, otherwise the operations removed from A' will no longer have the possibility of being processed at their current heads.

The reasoning above can be extended to the case where p' operators are available along $[T, C)$ and the number of machines in $R_{A'}$ is $m' \geq p'$. In this case A' can be reduced to maintain the operations of only $m' - p' + 1$ machines in order to guarantee that all the operations in A' have the opportunity to get scheduled at their heads in G_{SC} .

The set of operations obtained this way is termed B and it is clear that $|B| \leq |A'| \leq |A|$. An important property of this schedule generation scheme is that if the number of operators is large enough, in particular if $p \geq \min(n, m)$ so as $JSO(n, p)$ becomes $J||\Sigma C_i$, it is equivalent to the $G\&T$ algorithm. So, we call this new algorithm $OG\&T$ (Operators $G\&T$). From the reasoning above the following result can be established.

Theorem 1. *Let \mathcal{P} be a $JSO(n, p)$ instance. The set \mathcal{S} of schedules that can be obtained by the $OG\&T$ algorithm to \mathcal{P} is dominant, i.e. \mathcal{S} contains at least one optimal schedule.*

4 Search Algorithm

Our approach is an implementation of the best-first search strategy proposed by Nilsson [4]. The reason for choosing best-first search is that this strategy allows to fully exploit the dominance rules presented in section 4.3. This algorithm starts from an initial state and then in each step it expands the first one of the set of candidate states stored in the $OPEN$ list. The $OPEN$ list is sorted by non decreasing f -values, where f is the evaluation function for the states. $f(s)$ gives an estimation of the cost of the best schedule that can be reached from s , denoted as $f^*(s)$. In the following four subsections we describe the main components of the best-first search algorithm.

4.1 Search Space

The search space is derived from the $OG\&T$ schedule generation scheme. For a problem instance \mathcal{P} , in the initial state, none of the operations are scheduled yet and so it is defined by the disjunctive graph G . In intermediate states, a subset of operations SC is already scheduled. To obtain the successors of a state defined

by G_{SC} , the set B is built as indicated previously and then one successor state is generated from each operation u in B in which u is scheduled at its current head in G_{SC} .

We traverse this search space as a tree, even though it could be traversed as a graph as well. This is clear as two intermediate states with the same operations scheduled might represent the same subproblem and so this situation could be checked for avoiding duplications. However, as we will see below, the test defined to establish dominance relations among states is in fact a generalization of the procedure for checking duplications.

From theorem [1](#) above, it is clear that the search tree includes at least one optimal solution.

4.2 Heuristic Functions

We use the A^* version of best-first search, so the evaluation function is of the form $f(s) = g(s) + h(s)$, where $g(s)$ denotes the total flow time accumulated in the state s . Then, $h(s)$ is a heuristic function that estimates the additional cost required to reach a solution from s . We consider two admissible heuristics derived from problem relaxations.

The first one, termed h_{PS} , is borrowed from [6](#) where the problem $J||\Sigma C_i$ is considered: it relies on relaxing non-preemption and the capacity constraints of all the operators and all but one of the machines.

The second one, termed h_{OP} , is derived from a problem relaxation where non-preemption and the capacity constraints of the machines are relaxed.

Both h_{PS} and h_{OP} are computed in polynomial time and we take $h(s) = \max(h_{PS}(s), h_{OP}(s))$.

4.3 Dominance Rules

The effective search tree may be reduced by means of dominance relations among states similar to that exploited in [6](#) for the classic job-shop problem. Given two search states s_1 and s_2 , s_1 dominates s_2 iff $f^*(s_1) \leq f^*(s_2)$. In general, dominance relations cannot be easily established, but in some particular cases an efficient condition for dominance can be defined. For the search space above, a simple and effective dominance rule is defined as follows. If s_1 and s_2 are states having the same operations scheduled, SC , then s_1 dominates s_2 if the following three conditions hold, where $r_v(s)$ denotes the head of v in state s and $av(s)$ the availability of operators in this state:

1. $r_v(s_1) \leq r_v(s_2)$, for all $v \notin SC$.
2. $\sum_{\theta_{iv_i} \in SC} r_{iv_i}(s_1) \leq \sum_{\theta_{iv_i} \in SC} r_{iv_i}(s_2)$.
3. $av(s_1) \geq av(s_2)$.

From conditions (1) and (2) it is clear that the total flow time of the best schedule that can be obtained from s_1 is not greater than the total flow time of the best schedule reachable from s_2 , provided that the availability of operators in s_1 is not

worse than it is in s_2 . The availability of operators in a state can be evaluated as follows. Let $t_1 \leq \dots \leq t_p$ be the times at which the operators get idle in the state s (here it is worth noting that the operator available at time t_i is any O_j , $1 \leq j \leq p$). If u^* is the unscheduled operation with the lowest head in s , then none of the operators can get busy again before r_{u^*} , so we can consider that the operators are actually available for the unscheduled operations at times $t'_1 \leq \dots \leq t'_p$, where $t'_i = \max(r_{u^*}, t_i)$. So the availability of operators in state s is defined as the ordered vector $av(s) = (t'_1, \dots, t'_p)$ and $av(s_1) \leq av(s_2)$ iff $t'_{1i} \leq t'_{2i}$, $1 \leq i \leq p$.

The implementation of the dominance rules can be done as follows. When a state s is considered for expansion, s is compared to all the expanded states having the same operations scheduled. This can be done efficiently if the expanded states are stored in a CLOSED list implemented as a hash table where the key values are bit-vectors representing the scheduled operations.

Note that this pruning method generalizes the procedure for checking duplications as in these situations the nodes dominate each other.

4.4 Upper Bounds Calculation

Best-first search can be combined with a greedy algorithm to obtain an any-time algorithm in the following way: each time a state s is selected for expansion, the greedy algorithm is issued from s to obtain one of the solutions reachable from s . This way, approximate solutions are obtained from the beginning of the search and a number of states are pruned under the condition $f(s) \geq UB$, where UB is the best upper bound found so far. This process is time consuming, and one possibility to reduce the time taken is to issue the greedy algorithm just one from every 100 expansions or so.

We introduce upper bounds calculation in the following way: starting from the selected state s , the greedy algorithm traverses a branch of the search tree until a goal state is reached. In each iteration the set of candidate operations B is computed as done by the *OG&T* algorithm, then an operation is selected from B to be scheduled, in accordance with a heuristic estimation. We have opted to use the h -values for this purpose.

5 Computational Results

The purpose of the experimental study is to assess our proposal (BF) and to compare it with the IBM ILOG CPLEX CP Optimizer tool (CP), as no other approaches have been published up to date for the problem at hand. In CP, the p operators were modeled as a nonrenewable cumulative resource of capacity p .

We have experimented across two sets of instances. The first one includes a number of instances from the *OR*-library, in particular *FT06* (6 jobs \times 6 machines) and *LA01 – 05* (10 \times 5). For each instance, all values in the interval $[1, \min(n, m)]$ are considered as the number of operators p . The second benchmark is that proposed in [1], all these instances have $n = 3$ and $p = 2$

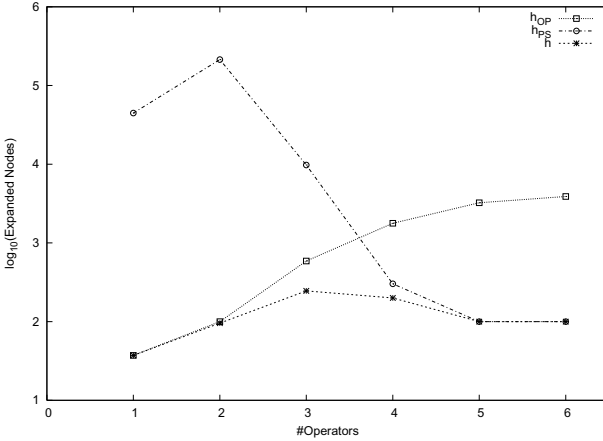


Fig. 2. Heuristics evaluation for instance FT06

and are characterized by the number of machines (m), the maximum number of operations per job (v_{max}) and the range of processing times (p_i). A set of small instances was generated combining three values of each parameter: $m = 3, 5, 7$; $v_{max} = 5, 7, 10$ and $p_i = [1, 10], [1, 50], [1, 100]$. Also, a set of larger instances was generated with $m = 3$, combining $v_{max} = 20, 25, 30$ and $p_i = [1, 50], [1, 100], [1, 200]$. In all cases, 10 instances were considered from each combination. The sets of small instances are identified by numbers from 1 to 27: set 1 corresponds to triplet 3 – 5 – 10, the second is 3 – 5 – 50 and so on. The sets of large instances are identified analogously by labels from L1 to L9.

In this study, we have given a time limit of 300 seconds. The target machine was Intel Core II (2,13 GHz), 2 GB RAM. and Windows XP (32 bit). The algorithms are coded in C++.

BF issues the greedy algorithm every 100 expanded nodes in all the experiments. This way, the algorithm becomes any-time and a number of nodes with $f(s) \geq UB$ can be pruned. Anyway, in our experiments we have observed that very few states get pruned by this condition.

In the first series of experiments, we have evaluated the heuristics h_{PS} and h_{OP} separately and in combination across 6 instances built from FT06 with p ranging from 1 to 6. The dominance rules described in section 4.3 were also used. The results are summarized in Figure 2. Here we report the number of expanded nodes (in a log₁₀ scale) for each combination of instance and heuristic. All the instances were solved to optimality by 300 s, with only one exception: $p = 2, h_{PS}$. The most relevant observation in this case is the complementarity of heuristics h_{PS} and h_{OP} . This is quite reasonable as h_{OP} is expected to be a better estimation than h_{PS} when the number of operators is small and the contrary can be expected when the number of operators is large. So, when they are used in combination to define $h(s) = \max(h_{PS}(s), h_{OP}(s))$, a synergetic effect is obtained as well. Also, it can be observed that the hardest instances

Table 1. Summary of results from Agnetis instances (300 s)

Instances	Opt.	CP			BF			
		T.(s)	#Sol.	%Err.	T.(s)	#Sol.	%Err.	
SMALL	1-9	506,14	0,03	90/90	0,00	0,01	90/90	0,00
	10-18	714,79	0,10	90/90	0,00	0,04	90/90	0,00
	19-27	1003,33	0,88	90/90	0,00	0,13	90/90	0,00
LARGE	L1-L3	4665,20	288,82	2/30	0,52	5,80	30/30	0,00
	L4-L6	5952,37	300	0/30	1,65	19,03	30/30	0,00
	L7-L9	7101,40	300	0/30	1,93	45,83	30/30	0,00

are those with intermediate values of p . There is also a correlation between the expanded nodes and the CPU time. Averaged for the 6 instances, it takes 56 s using h_{PS} (as it cannot solve the instance with $p = 2$ by 300 s), it takes 0,5 s with h_{OP} and less than 0,5 s with h . We will use h in all the remaining experiments.

We have also tried to assess the effectiveness of the pruning by dominance method across the instances from [1]. For this purpose, we solved the 270 small instances with $n = 3$ and $p = 2$. BF using the dominance rules needs to expand 75,66 nodes on average and 1182,87 nodes without using them. Considering the 90 large instances, BF solves all of them by 300 s using the dominance rules, and only 2 without using them. Hence, we will exploit the dominance rules in all the remaining experiments.

In order to compare BF and CP, we have considered the 360 instances proposed in [1]. Table 1 reports the results of these experiments, averaged for subsets of instances with the same number of operations per job v_{max} . We report the average optimal cost (Opt) and, for each algorithm, the time taken in seconds (T), the number of instances solved to optimality and proven to be optimal (#Sol) and the mean relative error in percentage w.r.t. the best lower bound given by the f -value of the last state expanded by BF (%Err). As we can observe BF is able to solve to optimality all the instances in much less time than 300 s, whereas CP can only solve the small instances and 2 of the 90 large instances.

Finally, Table 2 shows the results from instances LA01 – 05 averaged for the same number of operators (#Op). A remarkable observation is that CP cannot solve to optimality any of the 25 instances, and BF solves 16. We can also observe that the hardest instances are those with 3 and 4 operators. In these cases, BF reaches better solutions than CP.

From this experimental study we can conclude that the proposed best-first search algorithm (BF) clearly outperforms CP.

6 Conclusions

We have proposed an algorithm that combines best-first search and greedy search to solve the job-shop scheduling problem with operators. This algorithm is based on a new schedule generation scheme termed *OG&T*. The effectiveness of this

Table 2. Summary of results from instances *LA01 – 05* (300 s)

#Op.	CP			BF		
	T.(s)	#Sol.	%Err.	T.(s)	#Sol.	%Err.
1	300	0/5	0,01	0,20	5/5	0,00
2	300	0/5	2,04	29,80	5/5	0,00
3	300	0/5	4,56	276,40	2/5	1,29
4	300	0/5	4,00	300,00	0/5	3,38
5	300	0/5	0,26	167,80	4/5	0,36

algorithm relies on two heuristic estimations derived from problem relaxations and a method for pruning states based on dominance relations among states. We have reported results from an experimental study across conventional and new instances. This study shows that our approach outperforms the IBM ILOG CPLEX CP Optimizer constraint programming approach.

As future work we plan to experiment with heuristic search algorithms other than best-first, for example partially informed depth-first search as it was done in [5] for the classical job-shop problem, and to consider other variants of the problem more interesting from a practical point of view, focused on new operator constraints due to differences in the skills or time constraints due to labor rules.

Acknowledgments. We are grateful to the referees for their thoughtful comments and to Andrea Pacifici and Marta Flamini for making their benchmark instances available. This research has been supported by the Spanish Ministry of Science and Innovation under research project TIN2010-20976-C02-02 and by the Principality of Asturias under grant FICYT-BP09105.

References

1. Agnetis, A., Flamini, M., Nicosia, G., Pacifici, A.: A job-shop problem with one additional resource type. *J. Scheduling* 14(3), 225–237 (2011)
2. Giffler, B., Thompson, G.L.: Algorithms for solving production scheduling problems. *Operations Research* 8, 487–503 (1960)
3. González, M.A., Vela, C.R., Sierra, M.R., Varela, R.: Tabu search and genetic algorithm for scheduling with total flow time minimization. In: *COPLAS 2010*, pp. 33–41 (2010)
4. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Sys. Science and Cybernetics* 4(2), 100–107 (1968)
5. Mencía, C., Sierra, M.R., Varela, R.: Partially informed depth first search for the job shop problem. In: *Proceedings of ICAPS 2010*, pp. 113–120. AAAI Press (2010)
6. Sierra, M.R., Varela, R.: Pruning by dominance in best-first search for the job shop scheduling problem with total flow time. *Journal of Intelligent Manufacturing* 21(1), 111–119 (2010)

OntoMetaWorkflow: An Ontology for Representing Data and Users in Workflows

Alvaro E. Prieto, Adolfo Lozano-Tello, and José Luis Redondo-García

Quercus Software Engineering Group, University of Extremadura, av. Universidad s/n
10071, Spain
{aeprieto, alozano, jluisred}@unex.es

Abstract. Administrative processes are a type of business process commonly used in public institutions and large companies. These processes are generally characterized by involving the submission of some type of application form. This form must be considered at different stages by different users who need to handle current information of a dossier in order to provide new data in each activity until the end of the process. These processes are frequently reused because there are often similar processes within the organizations. The classification of the data managed by the activities and the categorization of the different users involved in every activity has great importance in these processes. The use of ontologies for modeling the workflows of administrative processes can provide significant improvements in this reuse process. In this paper, we describe OntoMetaWorkflow, a generic ontology to represent canonical workflow terms in the domain of administrative processes, and the methods that use it.

Keywords: business process, administrative process, workflows, ontologies, WEAPON.

1 Introduction

Administrative processes are generally used in administrative or legal ambits. They are characterized by being initiated by a user and which must be attended to or evaluated by other different users following a perfectly defined protocol for data, times and agents involved. Examples could be the management of public contest bids, loan application procedures or a simple holiday application.

These processes are often defined generically in the level of management or governance of the organization. However, these processes must be reused in the lower levels of the hierarchy dependent institutions, subdepartments, delegations, etc. in order to be applied in them. This reuse may become even more complex when the generic definition of the process is not done within the organization but rather it can be determined by rules enacted by an external organization and that rules are mandatory in order to use this process.

Workflow Management Systems (hereinafter referred to as WfMS) are applied to different types of business processes including administrative processes. Administrative processes do not usually require a complex WfMS with advanced

characteristics. In most cases, it can be more useful to have a WfMS with features that facilitate to share and reuse this type of process. Often, the reuse of this type of process only entails the change in the definition of the data structures managed in each activity or the users that can perform it. Because of this, the division of the definition of the workflow of the process into three separate but related definitions could be an important feature in order to reuse easily these processes. On the one hand, the definition of data structures to be managed by the process activities, on the other, the users that can perform each activity and, finally, the process activities together with the relationships among the three definitions.

For this reason, the use of ontologies as a basis for this type of WfMS could be very useful due to their characteristics of complete and precise representation of terms that make the integration of a data scheme easier and require less effort in reuse. Due to these characteristics, this type of WfMS will be easily reusable in similar institutions or companies. An appropriate case of application to reuse processes is the WfMS model based on ontologies was proposed in [1]. This model provided a generic ontology described in [1] as the basis of workflow representation, together with methods (and their respective software tools) to identify and to exploit the workflows of an administrative process. However, in some cases, the definition of the generic ontology did not allow easy reuse of process and the business flow. We have restructured its ontology to improve the reuse process. This restructuring reduces the time and effort in the reuse process of workflow activities, data and participants.

In this paper, a description of the new ontology, called *OntoMetaWorkflow*, is presented. The methods for defining administrative processes using *OntoMetaWorkflow* are also presented together with an example of the definition of a loan application process in ontologies. In addition to this, a brief description of the redefinition of the WfMS [1], now called *WEAPON* (Workflow Engine for Administrative Processes based on ontologies), is also presented.

This paper is structured as follows: section 2 identifies existing works that use ontologies in WfMS, section 3 describes the *WEAPON*, section 4 describes the ontology *OntoMetaWorkflow* and section 5 details the methods for defining and reusing administrative processes in ontologies using *OntoMetaWorkflow*.

2 Use of Ontologies in WfMS

The ontologies, applied to business process definition, provide a complete, precise and shared terminology about a particular domain which facilitates integration and which will be easily reusable by the same or another organization. These advantages provide a considerable saving of time and effort in processes and data definition tasks.

In particular, the application of ontologies to WfMS have been used previously in approaches as the one of Vieira et al. [2] that proposes a solution to make workflow execution more flexible and is possibly the first work integrating both fields. Also interesting is the work of Gasevic et al. [3] which provides a Petri net ontology. Haller et al. [4] present a multi meta-model process ontology, called *m3po*, which relates workflow models to choreography models. Finally, Abramowicz et al. [5] present a

semantically enhanced Business Process Modeling Notation [6], namely the sBPMN ontology. We can also mention the approaches presented in [7,8] as examples of integration of both fields and a recent survey about Semantic Business Process Management is available in [9].

Unlike the previous approaches, this paper presents an ontology for representing administrative processes together with their activities, domain data and users involved. Although several consolidated models and languages of workflow representation exist [10,11,12], the application of ontologies in this field, used directly or as a definition of a metalanguage, can provide the following advantages:

- The users, following methodologies for building ontologies, can obtain complete, precise and shared definitions of administrative process workflows.
- The domain data of a process or the users which participate in it can be changed without modifying the definition of the data managed by activities or the definition of the workflow.
- Workflow definitions, represented in ontologies, are more easily reusable although the reuse process may involve some effort, mainly in the processes of search, selection, and in some cases, adaptation to the new system. These factors are discussed in detail in [13].

3 WEAPON: Workflow Engine for Administrative Processes Based on Ontologies

WEAPON (Workflow Engine for Administrative Processes based on ONtologies), is a WfMS that proposes the use of ontologies to define and manage administrative processes and is more reusable than the first WfMS based on ontologies proposed in [1]. Basically, WEAPON proposes how a workflow designer must define, on one hand, the taxonomy of relevant data of the domain and the taxonomy of users which can participate in the workflow and, on the other hand, the activities that the process contains together with the identification of which type of user defined can perform them and the data managed by every activity. This ensures that the processes are well defined and more reusable and, in addition, the classes of domain data and the classes of workflow participants can be modified without changing the representation of the workflow process. Moreover, it should be noted that the taxonomies of classes and instances that may be needed as domain data or workflow participants, can be defined in ontologies in the organization itself or can be reused from ontology repositories.

The architecture of WEAPON presents a series of interrelated components (see Fig. 1). These components are:

1. OntoMetaWorkflow, the ontology, represented in OWL Language, for the generic definition of workflows.
2. OntoDD, an ontology of the domain data and workflow participants built following the specifications of OntoMetaWorkflow.
3. OntoWF, an ontology of the workflow of the administrative process built following the specifications of OntoMetaWorkflow and OntoDD.

4. WEAPON Designer, is the tool that allows users to combine WF-Net [14] representation with OntoMetaWorkflow and the OntoDD of a domain in order to design the OntoWF Ontology for a specific administrative process.
5. WEAPON Manager, is the web application that reads OntoDD and OntoWF ontologies and generates the web forms and the database that manage the workflow of the administrative process.

It is important to note that, due to the activities of this type of process being tightly bound to the data managed, ideas of traditional WfMS as well as some provided by the Case Handling approach [15] have been used in WEAPON. Basically, Case Handling is a data-driven approach where each activity is associated with at least one data object of the process managed. Moreover, it proposes the use of forms, associated to activities, for managing data. WEAPON uses a similar idea but with the difference that the activities are not associated to predefined forms but rather that the forms are built dynamically for each activity using the definitions contained in OntoDD and OntoWF. This provides advantages for independence and reuse of the different representations.

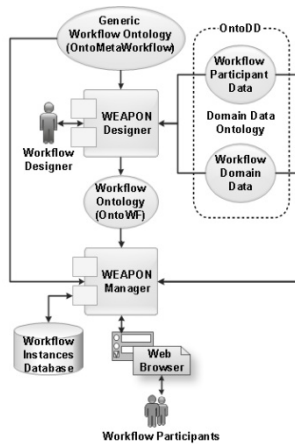


Fig. 1. Architecture of the WEAPON WfMS

4 OntoMetaWorkflow: A Generic Ontology for the Representation and Reuse of Workflows

OntoMetaWorkflow¹ contains the terms that form the workflows of administrative processes and their relationships. This ontology is built adapting the definitions of workflow elements provided by the WfMC [16] to the specific characteristics of administrative processes. It has been developed following the METHONTOLOGY

¹ <http://quercusseg.unex.es/weapon/?download=OntoMetaWorkflow.owl>

methodology [17]. The different representation elements of OntoMetaWorkflow are classified into two types:

1. Definition elements of OntoDD: are used to define the classes and properties that represent the common data and the potential users of all similar processes within a domain. These elements are the *Domain Data*, *Workflow Participant* and *Root* classes. *Domain Data* stores common data of all instances of an administrative process and has the *External Document* and *Location* attributes. The *Workflow Participant* class stores the users involved in the process and has Id, Password, Name, Surname and Email attributes. The *Root* subclass is a special class that can administer the WEAPON Manager WfMS.
2. Definition elements of OntoWF: are used to define the classes and properties that represent a particular process, that is, the sequential flow of activities and their relationships with the elements of the domain defined in OntoDD. These elements are the *Administrative Process* and *Activity* classes. *Administrative Process* class is used for representing the process managed by the WfMS and has defined the *Generated By* relationship. The *Activity* class represents a logical unit of work and has defined the *Is Performed By* and *Before* relationships and the *Before Control Flow Pattern*, *Select Class Of Domain Data*, *Show Class of Domain Data*, *Select Instance Of Domain Data*, *Show Instance of Domain Data*, *Fill In Instance Attributes of Process*, *Show Instances Attribute*, *Days Time Frame*, *Day Notice* and *Activity Description* attributes.

A graphical representation of OntoMetaWorkflow is available in².

5 Methods for Defining and Reusing Administrative Processes Using OntoMetaWorkflow

It is necessary to apply two methods to define administrative processes as ontologies using OntoMetaWorkflow. The first method is for building the domain ontology, OntoDD. The second one is for building the ontology of the process, OntoWF. Both methods together with an example of how to build OntoDD and OntoWF of a loan application process are explained in detail in the following subsections.

In this example (shown in WEAPON Designer in Fig. 2), it is necessary to know, among other domain data, the types of credit offered by a bank or the internal classification of risk factor applied by the bank to the applicants. The workflow designer could reuse an existing ontology about types of credit or risk factors with a consequent saving of time. Moreover, if we suppose that in this example the rules for the processes of loan applications are established in a generic workflow by the Central Bank of the country, then, it would be enough to adapt this generic definition to every bank in particular, simply adding its taxonomy of domain data (credit types or risk factor) and its taxonomy of users (customer, credit analyst, etc.).

² <http://quercusseg.unex.es/weapon/?OntoMetaWorkflow>

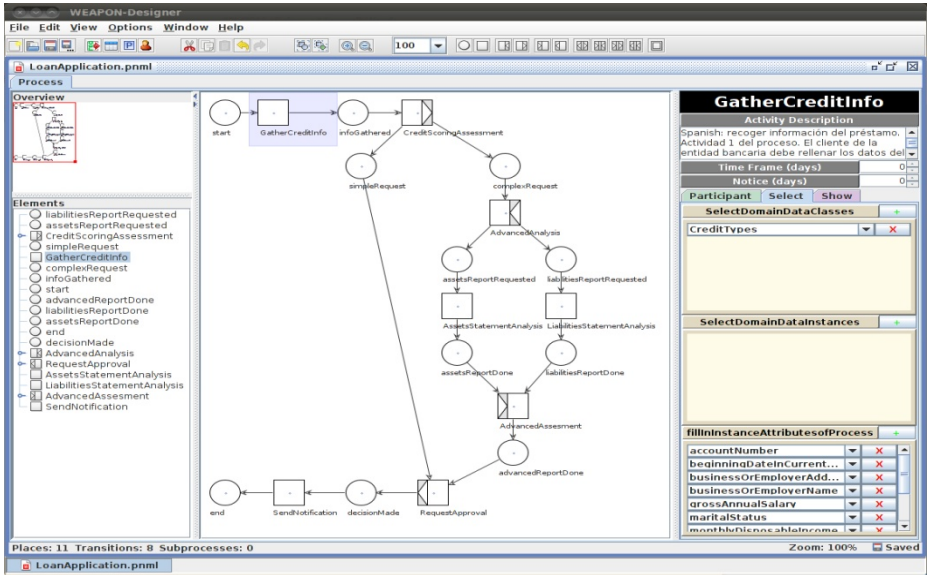


Fig. 2. Loan application process in WEAPON Designer

5.1 OntoDD: Ontology of Relevant Data of an Specific Domain

OntoDD imports the concepts defined in OntoMetaWorkflow and must contain, firstly, the taxonomy of data which will be used in the corresponding domain and, secondly, the taxonomy of the possible workflow participants. The main rule which must be fulfilled by OntoDD is that the root elements of each one of its taxonomies is linked with the superclasses defined in OntoMetaWorkflow, *Domain Data* and *Workflow Participant* (or *Root* if the user is a WfMS administrator) respectively.

The development of this ontology can be carried out in a simple way, with some application for building ontologies in OWL like Protégé, carrying out the following steps:

1. Import OntoMetaWorkflow so its elements will be superclasses in OntoDD.
2. Identify those common data taxonomies of the domain in order to define them as subclass of *Domain Data*. That is, the taxonomies of data that will be common to all instances of the administrative process must be defined here. In the loan application process, taxonomies such as *Credit Types* with subclasses as *Loan* or *Mortgage* or *Risk Factor* with subclasses as *High Risk* or *Low Risk* will be subclass of *Domain Data*.
3. Define the common properties of each taxonomy of *Domain Data*. In particular, in the loan application process, the *Credits Types* will be classified by having different *terms* or *interest rate ranges* while the *Risk Factor* of a loan application process will be defined by means of the result obtained in a *risk assessment test*.

4. Set the property values for every subclass of the *Domain Data* and, if exist, for the instances of the subclasses. In the loan application process, the taxonomies defined as domain data do not contain instances. Thus, it is only necessary to set property values for subclasses. Typical of an administrative process with instances in the *Domain Data* is an incident management process. In this case, a possible subclass of *Domain Data* as *Involved Material* will contain instances of every printer or workstation that the company may have.
5. Identify the different type of users that will participate in the workflow as subclasses of *Workflow Participant*. In the loan application process, there will be classes such as *Customer*, *Credit Analyst* or *Credit Manager*. This step also includes define at least one of these classes as subclass of *Root* (*Credit Manager* in the loan application process), create the instances of every real user and set the values *Id*, *Password*, *Name*, *Surname* and *Email* properties for every instance.

The product of applying this method is an ontology with the domain data that can be used by the instances of the workflow and the users which can carry out it. As example, the OntoDD ontology, for the loan application domain, is available³.

Reuse in OntoDD. Existing domain ontologies can be reused in OntoDD adding them as subclass of *Domain Data*. Ontologies with the users of the organization can be reused as *Workflow Participant*. In this case it is necessary to take the next simple actions:

1. Set every type of user in the ontology as subclass of *Workflow Participant*.
2. It is necessary to define some of the classes as subclass of *Root*.
3. For every instance of reused classes is necessary to set the value for *Id*, *Password*, *Name*, *Surname* and *Email* definition attributes.

5.2 OntoWF: Ontology of the Workflow of the Specific Administrative Process

This ontology represents the workflow of the corresponding administrative process that will be managed by the WfMS. This ontology contains the concrete workflow of the administrative process, including its properties, the activities that it contains, the order of execution of said activities, the relevant data of OntoDD that will be shown or modified in an activity and the participants which can carry out every activity.

WEAPON Designer is used in the design of the OntoWF Ontology for a specific administrative process. However, it is possible to build OntoWF manually without the use of this tool, with a large effort. In any case, these steps must be followed:

1. Import the OntoDD defined in the domain and OntoMetaWorkflow.
2. Define the process to manage as subclass of *AdministrativeProcess*. A class called *Loan Application* will be subclass of *AdministrativeProcess* in our example.

³ http://quercusseg.unex.es/weapon/?download=OntoDD_LoanApplication.owl

3. Define the properties of the process. These properties will be displayed or modified for each instance of the process. Personal and financial data of the applicant or the different reports that will be made during the process will be properties of each instance of the *Loan Application* class.
4. Indicate what subclasses of *Workflow Participant* can create instances of the process using the Only universal restriction on *Generated By* definition relationship. In the loan Application process, only a *Customer* can apply for a loan.
5. Define each activity of the process as subclass of *Activity*. In the loan application process, there will be activities as *Gather Credit Info* or *Request Approval*.
6. Additionally, for each subclass of *Activity* it is necessary to:
 - (a) indicate which activities precede it using *Before* relationship,
 - (b) if the *Before* relationship is restricted to two or more activities, then indicate whether these activities have been carried out in parallel or conditionally. The *Before Control Flow Pattern* attribute will take the value *and* in the first case and *xor* in the second one. This attribute cannot take both values in the same activity,
 - (c) indicate what subclasses of *Workflow Participant* can carry out the activity using the *Is Performed By* relationship. This relationship should always be restricted to at least one subclass of *Workflow Participant*,
 - (d) if is necessary to choose among different subclasses of *Domain Data*, then indicate the name of the root class of these subclasses using *Select Class Of Domain Data* attribute,
 - (e) if is necessary to show some subclasses of *Domain Data* that has been selected in a previous activity, then indicate the name of the root class of these subclasses using the *Show Class Of Domain Data* attribute,
 - (f) if is necessary to choose among different instances of *Domain Data* subclasses, then indicate the name of the root class of the subclasses that contain the instances using the *Select Instance Of Domain Data* attribute,
 - (g) if is necessary to show some instances of *Domain Data* subclasses that has been selected in a previous activity, then indicate the name of the subclasses that contain the instances using the *Show Instance Of Domain Data* attribute,
 - (h) indicate the process properties (defined in step 3 of this method) than can be filled in, or modified, using the *Fill In Instance Attributes of Process* attribute,
 - (i) indicate the process properties (defined in step 3 of this method) than can be displayed, using the *Show Instance Attributes of Process* attribute,
 - (j) set the number of days for doing the activity using *Days Time Frame* attribute,
 - (k) indicate, using the *Day Notice* attribute, the number of days before the deadline of the activity in which the person responsible for the activity will be warned in order to finish,
 - (l) explain, in natural language, the actions to do in the activity using the *Activity Description* attribute.

The product of applying this method is an ontology with the definition of the workflow of the corresponding process. This definition contains its properties and its activities together with all the relationships among activities, domain data, process

properties and workflow participants that are necessary to carry out it. As example, the OntoWF ontology, for the loan application process, is available⁴.

Reuse in OntoWF. The reuse in OntoWF is derived from the ease of reuse in OntoDD and how the activities in OntoWF use the OntoDD elements. Following the loan application example, a customer can choose among different credits because of *Gather Credit Info* activity has a Value restriction on *Select Class Of Domain Data* definition attribute with *Credit Types*. If other departments or other banks want to reuse this process with their own credits, it only must change the subclasses of *Credit Types* in the original OntoDD by its own credits subclasses and the process works on. This idea also applies if it is necessary to change the *Workflow Participants* of the process.

6 Conclusions

In administrative processes, the classification of the data managed by the activities and the categorization of the participant users in every activity has great importance because they are reusable terms in similar processes within the organizations. Due to this, the representation of workflows of administrative processes using ontologies provide significant advantages such as ease of use and information which is complete, consistent and shared both in data and processes.

To establish the generic concepts that are used in the definitions of workflows, we have presented OntoMetaWorkflow, an ontology which specifies the elements and rules that define workflows according to the standards and recommendations of the WfMC. Using this ontology, it is possible to represent the relevant data of the corresponding domain and the users involved in each particular administrative process or simple business process. The methods of WEAPON have also been presented in this paper.

OntoMetaWorkflow and the methods of WEAPON have been tested in several domains, mainly in administrative processes of University of Extremadura. They are not designed for being used with processes that need complex queries to database, internal calculations or the use of external applications. However, they work properly with administrative processes that are fully oriented to humans and, specially, in those processes that involve submitting some type of application to be considered at different stages, where different participants need to handle current information of a dossier in order to provide new data in the corresponding activity.

Although OntoMetaWorkflow is the basis of the WEAPON WfMS, it has been designed with the intention that can be reused in other projects or with other tools by those researchers interested in using ontologies to manage workflows.

Acknowledgments. This work has been developed under support of Ministerio de Ciencia e Innovacion Project (TIN2008-02985), FEDER, Junta de Extremadura and

⁴http://quercusseg.unex.es/weapon/?download=OntoWF_LoanApplication.owl

Plan de Iniciación a la Investigación, Desarrollo Tecnológico e Innovación 2010 de la Universidad de Extremadura (ACCVII-04).

References

1. Prieto, A.E., Lozano-Tello, A.: Use of Ontologies as Representation Support of Workflows Oriented to Administrative Management. *J. Netw. Syst. Manag.* 17(3), 309–325 (2009)
2. Vieira, T.A.S.C., Casanova, M.A., Ferrão, L.G.: On the design of ontology-driven workflow flexibilization mechanisms. *J. Braz. Comp. Soc.* 11(2), 33–43 (2006)
3. Gasevic, D., Devedzic, V.: Petri net ontology. *Knowl.-Based Syst.* 19(4), 220–234 (2006)
4. Haller, A., Oren, E., Kotinurmi, P.: m3po: An Ontology to Relate Choreographies to Workflow Models. In: 3rd IEEE International Conference on Services Computing, pp. 19–27. IEEE Computer Society, Los Alamitos (2006)
5. Abramowicz, W., Filipowska, A., Kaczmarek, M., Kaczmarek, T.: Semantically enhanced Business Process Modelling Notation. In: 2nd Workshop on Semantic Business Process and Product Lifecycle Management, CEUR-WS, Innsbruck, Austria, pp. 88–91 (2007)
6. OMG: Business Process Model and Notation (BPMN) 1.2 (2009)
7. Vidal, J., Lama, M., Bugarín, A.: A Workflow Modeling Framework Enhanced with Problem-Solving Knowledge. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4253, pp. 623–632. Springer, Heidelberg (2006)
8. Dang, J., Hedayati, A., Hampel, K., Toklu, C.: An ontological knowledge framework for adaptive medical workflow. *J. Biomed. Inform.* 41(5), 829–836 (2008)
9. Hoang, H.H., Tran, P.C., Le, T.M.: State of the Art of Semantic Business Process Management: An Investigation on Approaches for Business-to-Business Integration. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) Intelligent Information and Database Systems. LNCS, vol. 5991, pp. 154–165. Springer, Heidelberg (2010)
10. WfMC: Process Definition Interface – XML Process Definition Language Document No. WfMC-TC-1025, 2.1a (2008)
11. Aalst, W.M.P.V.D., Hofstede, A.H.M.T.: YAWL: yet another workflow language. *Inform. Syst.* 30(4), 245–275 (2005)
12. OASIS: Web Services Business Process Execution Language Version 2.0 (2007)
13. Lozano-Tello, A., Gómez-Pérez, A.: Applying the ONTOMETRIC Method to Measure the Suitability of Ontologies. In: Green, P.F., Rosemann, M. (eds.) Business Systems Analysis with Ontologies, pp. 249–269. IGI Global, Hershey (2005)
14. Aalst, W.M.P.V.D., Hee, K.V.: Workflow Management - Models, Methods and Systems. MIT Press, Cambridge (2002)
15. Aalst, W.M.P.V.D., Weske, M.: Case handling: a new paradigm for business process support. *Data Knowl. Eng.* 53(2), 129–162 (2005)
16. Hollingsworth, D.: The Workflow Reference Model. Document Number TC00-1003 Document Status - Issue 1.1 (1995)
17. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. Springer, London (2004)

Architecture for the Use of Synergies between Knowledge Engineering and Requirements Engineering

José del Sagrado, Isabel M. del Águila, and Francisco J. Orellana

Dpt. Languages and Computation,
Ctra Sacramento s/n, 04120 University of Almería, Spain
{jsagrado, imaguila, fjorella}@ual.es

Abstract. The application of Artificial Intelligence techniques in the processes of Software Engineering is achieving good results in those activities that require the use of expert knowledge. Within Software Engineering, the activities related to requirements become a suitable target for these techniques, since a good or bad execution of these tasks has a strong impact in the quality of the final software product. Hence, a tool to support the decision makers during these activities is highly desired. This work presents a three-layer architecture, which provides a seamless integration between Knowledge Engineering and Requirement Engineering. The architecture is instantiated into a CARE (Computer-Aided Engineering Requirement) tool that integrates some Artificial Intelligence techniques: Requisites, a Bayesian network used to validate the specification of the requirements of a project, and metaheuristic techniques (simulated annealing, genetic algorithm and an ant colony system) to the selection of the requirements that have to be included into the final software product.

Keywords: Requirement management, bayesian network, computer aided requirement engineering.

1 Introduction

The software development has been supported by Artificial Intelligence (AI) techniques for more than 20 years, since the appearance of the first intelligent editors. Currently we are witnessing a reemergence of AI and Software Engineering (SE), which has become a valid and potentially very valuable research field [15]. Expert knowledge is involved in every software development project since developers must face numerous decision tasks during requirements, analysis, design, and implementation stages. Therefore, if expert knowledge could be properly modelled and incorporated in the different processes of software development as well as in the CASE tools that support these processes, that would mean a great advantage for any software development.

Requirements stage is considered a good application domain for AI techniques because of requirements nature. Software requirements express and establish the

needs and constraints that contribute to the solution of a real world problem [11]. However, requirements tend to be imprecise, incomplete and ambiguous. In SE it is well known that this area is quite different from others because requirements reside in the problem space, whereas other software artifacts reside in the solution space [5]. Statistical studies and all the Chaos Reports [8,4], published since 1994, point out that tasks related to requirements are the main cause of disaster of software products. When requirement-related tasks are poorly defined or executed, the software product is typically unsatisfactory [22,3,4]. The role played by requirements is essential, as they are the basis for the analysis, design and implementation of the final product. Therefore, any improvement in the requirements stage will favorably affect the whole software life cycle.

Bayesian Networks [17,9,10], have been successfully applied with the purpose of enhancing specific activities related to SE knowledge areas. They have been applied in maintenance [14], defect and effort prediction [7,18,19] or implementation of a software project [12]. In addition, Bayesian networks have been successfully applied in Requirement Engineering (RE) [2], specifically in the prediction of the need for a review of the requirements' document [20].

Other AI techniques that have also been used to enhance the requirement stage are metaheuristic optimization techniques. Specifically, they have been used in the selection of the set of requirements that will be included in the development of a final software product [21].

Therefore, we need a seamless integration of RE and AI techniques to exploit the benefits of collaboration between these two knowledge areas.

By other hand, the biggest breakthrough in requirement management is when you stop thinking of documents and start thinking about information. Moreover, to be able to handle this information you have to resort to databases, particularly documental databases that have evolved into what nowadays are called CARE (Computer-Aided Engineering Requirement) tools. Among this type of tools the best known are the IRqA, Telelogic DOORS, Borland Caliber, and the IBM-Rational Requisite Pro. InSCo Requisite is an academic web CARE tool, developed by DKSE group at the University of Almería, which aids during the requirement development stage [16].

This work presents the architecture for the seamless integration of a CARE tool to manage requirements (i.e. InSCo Requisite) with some AI techniques (i.e. Bayesian networks and metaheuristics). Specifically, a Bayesian network, called Requisites [20], is used in the requirement validation task in order to validate the Software Requirements Specification (SRS) of a software development project. Metaheuristic techniques (Simulated Annealing, Genetic Algorithms and Ant Colony Systems) are used in the problem of selecting the subset of requirements among a whole set of candidate requirements proposed by stakeholders, that will be included in the development of a final software product [21].

The rest of this paper is structured as follows. Section 2 depicts the requirement engineering workflow. The architecture for the use of synergies between Knowledge Engineering and Requirements Engineering is explained in Section 3. Finally, the conclusions and future works are exposed in Section 4.

2 Enhancement of Requirement Engineering Workflow

The workflow depicted in Figure 1 shows an organization of the tasks that must be done in a software development project during Requirement Engineering stage. This workflow unifies the main classics methodological approaches [22,113], starts with a feasibility study that is built in order to determine the project scope and the availability of resources. Once the feasibility report is available, elicitation and analysis, specification and validation tasks are executed in an iterative way.

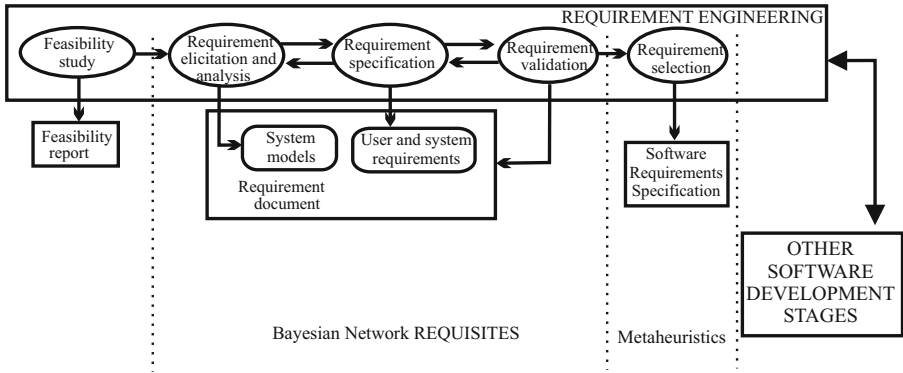


Fig. 1. Requirement Engineering workflow

Requirements are elicited or gathered from users through interviews and other techniques such as questionnaires or brainstorming. Usually this is a complex task because activities requiring human communication imply problems of understanding, and requirements have to be conceived without ambiguities in order to define what the system is expected to do.

In the next task, requirement specification, captured requirements are gathered in a document or its electronic equivalent, known as Software Requirements Specification (SRS). Early approaches to address this activity using computers involved the use of word processors. This way of supporting requirement specification can be tedious and prone to error for the management and maintenance of large sets of requirements. CARE tools appeared to give a solution to this problem, providing environments that make use of databases, allowing an effective management of the requirements of any software project.

Requirements validation is a task performed in order to check whether the elicited and specified requirements present inconsistencies, the information is incomplete or there are ambiguities in the system definition. The Bayesian network Requisites has been built, through interaction with experts and using several information sources, such as standards and reports, to support validation and specification partially. The aim of this network is to provide developers an aid, under the form of a probabilistic advice, helping them at the time of making

a decision about the stability of the current requirements specification. Requisites provides an estimation of the degree of revision for a given requirements specification. Thus, it helps the process of identify if requirements specification is stable and does not need further revision.

The elicitation-analysis-specification-validation cycle is carried out through several iterations in order to correct the defects found, and decide if the requirements specification has been completed in order to move towards the next task.

Finally, requirements selection task has as main objective to choose, from all the requirements defined in the specification, the subset of requirements that will be implemented. This selection is necessary due to the limitations of resources that usually appear in the feasibility report, and prevents development of all defined requirements. Requirements selection is a complex problem where many factors are involved (requirements priorities, development costs, customers' priorities, etc). The goal is to select a subset of requirements by searching for a set of requirements, which maximize satisfaction and minimize development effort considering the project constraints. This problem has been addressed in the literature using techniques from Artificial Intelligence, specifically metaheuristic algorithms [21] which have shown a performance similar to that exhibited by experts at the time of selecting the set of requirements to be developed in the software product.

3 Seamless Synergic Architecture

AI techniques described in this paper have demonstrated to obtain interesting results through different tests data [20,21]. However, it is difficult to put them in practice in real software projects. We strongly believe that having these AI techniques available in a CARE tool would be considerably helpful for any development team, making them more accessible even for non-expert people.

InSCo Requisite [16] is a web-based tool developed by DKSE research group at the University of Almería, to manage requirements of software development projects. It provides basic functionality allowing groups of stakeholders to work in collaboration. The fact of having the possibility of make changes to the tool, give us an exceptional opportunity to afford the integration of AI techniques in a CARE tool. This integration cannot be done in a straightforward way, because AI techniques and the CARE tools have been developed independently of each other. Therefore, it is necessary to define a communication interface between them preserving the independent evolution of both areas and achieving a synergic benefic effect between them.

This seamless synergic architecture is shown in Figure 2. The architectural pattern distinguish between three logically separated processes: the presentation (i.e. interface layer), the application processing (i.e. service layer), and the data management (i.e. data layer). This pattern (see upper picture in Fig. 2) provides a framework to create a seamless synergy, between knowledge-based tools and computer aided software engineering tools, at service layer becoming a more flexible management of interface and data.

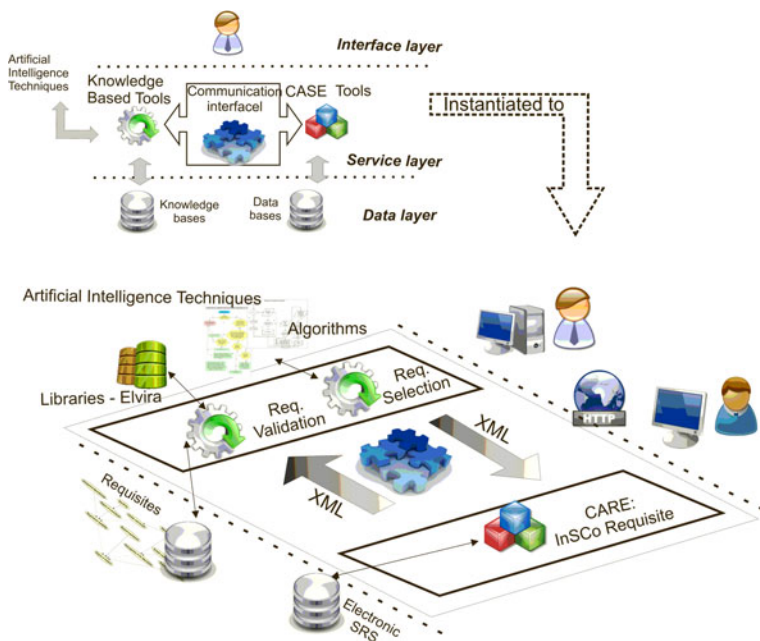


Fig. 2. Seamless synergic architecture

The architectural pattern can be instantiated for the enhanced requirement engineering workflow (see the bottom picture in Fig. 2). Interface layer represents the part of the architecture that provides users a mechanism to interact with the system. In our case, the interface is a web environment accessed from a web browser. Data layer is in charge of storing and managing the electronic representation of SRS handled by InSCo Requisite tool and the knowledge base that contains the Bayesian network Requisites. Service layer contains all the functionality provided by the overall system. The layer is composed by the CARE tool (i.e. InSCo Requisite), the AI techniques used to address requirements validation (i.e. Bayesian network Requisites) and requirements selection (i.e. metaheuristic algorithms) tasks, and a communication interface used to send and retrieve information between them.

The CARE tool is in charge of the management of all the information related to the development project (requirements, customers, etc) which is stored in a database. The knowledge-based tools carry out requirements validation and requirements selection tasks. Communication interface connect CARE and knowledge-based tools passing the required information needed for the execution of the appropriated processes. Thus, requirement validation receives metrics on the SRS and returns an estimation of the degree of revision for SRS; requirement selection receives resources effort bound and specific measures on individual requirements and identifies the set of requirements selected for implementations. All of these communication processes are performed through XML files. Next

subsections explain the AI techniques applied in requirement validation and requirement selection.

3.1 Requirements Validation Module

Bayesian networks [17,9] are a well-known Artificial Intelligence technique suitable in handling decision-making problems involving uncertainty. They have the advantage of having rich semantics and can be interpreted by the stakeholders without a high background on Statistics. From user's point of view, Bayesian networks provide a natural framework for relevance analysis and prediction tasks. Therefore, a Bayesian network can be used in requirement validation as a predictor that determines whether a requirements specification has enough quality to be considered as a baseline of a software project, establishing a contractual agreement between customers and developers about what is needed to be developed [20]. Bayesian network Requisites (see Figure 3) has been designed through interactions with experts and using several information sources, such as standards and reports, to tell us whether we can stop the iterations needed in order to define the SRS. To assess the goodness of a SRS, Requisites uses the following variables:

- *Stakeholders' expertise*: Degree of familiarity with expertise respect to tasks related to RE. Previous experience would lead to commit fewer errors.
- *Domain expertise*: Level of knowledge reached by the development team about the project domain. If developers and other stakeholders handle the same terminology, communication will be more effective.
- *Reused requirements*: If the number of requirements from reusable libraries is high, the overall specification of the requirements may not need new iterations.
- *Unexpected dependencies*: Unexpected dependencies between requirements usually involve a new revision of the specification of the requirements.
- *Specificity*: Number of requirements sharing the same meaning for all stakeholders. A higher specificity implies less revision and a shorter process of negotiation in order to reach a commitment.
- *Unclear cost/benefit*: Requirements included by stakeholders or developers whose benefits cannot be clearly quantified.
- *Degree of commitment*: Number of requirements that required a negotiation in order to be accepted.
- *Homogeneity of the description*: A good requirement specification must be described using the same level of detail. If there is no homogeneity, the specification will need to be revised.
- *Requirement completeness*: Indicates if all significant requirements are elicited and/or specified.
- *Requirement variability*: Represent that requirements have suffered changes. When a requirement specification changes, it needs an additional revision.
- *Degree of revision*: Value predicted by Requisites Bayesian network.

The structure of Requisites models the dependence relationships between variables. Note that each node in the network has attached its own conditional probability distribution given its parents. In the qualitative structure of Requisites, specificity, unexpected dependencies, reused requisites, stakeholder's expertise and domain expertise are not affected by any other variables. Degree of commitment and unclear cost benefit are related because if the degree of commitment (i.e. the number of requirements that have to be agreed) increases, then the level of specificity will be low. If stakeholders have little experience in the processes of RE, then it is more likely to lead to requirements which are unclear in terms of cost/benefits.

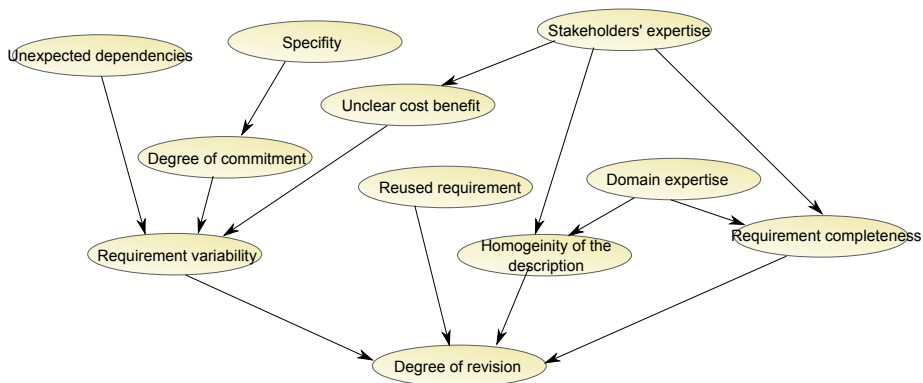


Fig. 3. Requisites Bayesian network [20]

The requirement completeness and homogeneity of the description are influenced by the experience of software and requirement engineers in the domain of the project and by stakeholders in the processes or tasks of RE. If experience is high, the specification will be complete and homogeneous because developers have been able to describe the requirements with the same level of detail and have discovered all requirements.

Requirement variability represents the number of changing requirements. A change in the requirements will be more likely to occur: if unexpected dependencies are discovered, if there are requirements that do not add any value to the end software, if there are missing requirements or if requirements have to be negotiated.

Bayesian network Requisites makes its prediction in order to indicate whether a requirement specification is sufficiently accurate or require further revision, performing an inference process in which some evidences or observations are used to calculate the marginal probability distribution of the variable degree of revision. The evidences (i.e. variables observed) are provided by the CARE tool that is in charge of extracting values of variables from the data about projects, requirements, users's activity and so on. Evidence is transferred from the CARE tool to Requisites through the communication interface as a XML

file. The Bayesian network Requisites is implemented using Elvira [6], a Java software package for the creation and evaluation of Bayesian networks. Then, an inference process is launched on Requisites computing the posterior marginal probability distribution of the unobserved variables given evidence. The results are sent back to the CARE tool via the communication interface as other XML file.

The main advantage of applying this architecture, besides the synergy between AI and RE, resides in the fact that the Bayesian network model can be modified without affecting the CARE tool and vice versa, providing a great flexibility.

3.2 Requirements Selection Module

The use of meta-heuristics techniques can help experts who must decide which is the set of requirements that has to be considered in the next development stages when they face contradictory goals. The main aim is to combine computational intelligence and the knowledge experience of the human experts with the idea of obtaining a better requirements selection than that produced by developer's judgment alone.

The selection of a set of requirements between all those previously defined and validated can be addressed using metaheuristic optimization techniques. Specifically, simulated annealing, genetic algorithms and an ant colony system [21] have been adapted to solve this problem. In order to work these metaheuristics algorithms for requirement selection need: a representation of the problem, which is amenable to symbolic manipulation, a fitness function based on this representation and a set of manipulation operators. InSCo Requisite generates an interface file in XML format containing all data needed for the execution of each of the metaheuristic algorithms. This file is transferred through the communication interface. Note that input adopts the same format for each of the algorithms, so it would be a simple task to add new algorithms. Each algorithm searches for a subset of requirements which maximize the customers satisfaction and minimize the required implementation effort within the given project constraints (see [21] for details). After its execution, the set of selected requirements obtained is sent as an XML file through the communication interface and is presented in the interface of InSCo Requisite. In this way, developers receive a feedback when perform the task of requirement selection.

4 Conclusions

The purpose of this work is to define a three-layer architecture with two objectives. On the one hand, allow the seamless collaboration between Requirement Engineering tasks and some Artificial Intelligence techniques in order to perform a software development project. And on the other, facilitate their parallel and independent evolution.

During a software development project, the tasks belonging to the requirements stage workflow are inherently difficult and uncertain, and are considered

a good domain for the application of AI techniques. In this work, we have structured the requirement workflow into several tasks, paying special attention to those tasks that can be enhanced or supported by knowledge-based techniques: requirement validation and requirement selection.

The generic seamless architecture has been instantiated taking advantage of the synergy between requirement management tools (InSCo Requisite), Bayesian networks (Requisites) and metaheuristic algorithms (Simulated Annealing, Genetic Algorithms and Ant Colony Systems). In this architecture, the communication interface is responsible for making the connection between CARE and knowledge-based tools, and has to pass the information required and needed for the execution of the appropriated processes for requirement validation and requirement selection tasks.

In the next future, we plan to instantiate our seamless synergic architecture to other Software Engineering stages (e.g. software maintenance or project management) by adding other knowledge models already developed in order to enhance these development stages. Also, we plan to enhance and automate the definition of the communication interface by defining languages that support it.

Acknowledgments. This work was supported by the Spanish Ministry of Science and Innovation under project TIN2010-20900-C04-02 and by the Junta of Andalucía under project TEP-06174.

References

1. Abran, A., Moore, J., Bourque, P., Dupuis, R., Tripp, L.: Guide to the Software Engineering Body of Knowledge 2004 Version. IEEE Computer Society, Los Alamitos (2004)
2. Barry, P.S., Laskey, K.B.: An Application of Uncertain Reasoning to Requirements Engineering. In: 15th Conference on Uncertainty in Artificial Intelligence, pp. 41–48. Morgan Kaufmann, Stockholm (1999)
3. Standish Group: Chaos Report. Technical report, Standish Group International (1994)
4. Johnson, J.: CHAOS chronicles v3.0. Technical report, Standish Group International (2003)
5. Cheng, B.H., Atlee, J.M.: Research directions in requirements engineering. In: Future of Software Engineering, FOSE 2007, pp. 285–303. Institute of Electrical and Electronics Engineers, Minneapolis (2007)
6. Elvira Consortium: Elvira: An environment for probabilistic graphical models. In: First International Workshop on Probabilistic Graphical Models (PGM 2002), Cuenca, España, pp. 222–230 (2002), <http://leo.ugr.es/elvira/>
7. Fenton, N., Neil, M., Marsh, W., Hearty, P., Marquez, D., Krause, P., Mishra, R.: Predicting software defects in varying development lifecycles using Bayesian nets. *Information and Software Technology* 49(1), 32–43 (2007)
8. Glass, A.R.L.: Facts and Fallacies of Software Engineering. Pearson Education, Inc., Boston (2002)
9. Jensen, F.V.: Bayesian Networks and decision graphs. Springer, New York (2001)
10. Jensen, F.V., Nielsen, T.: Bayesian networks and decision graphs. Springer, New York (2007)

11. Kotonya, G., Sommerville, I.: *Requirements Engineering: Processes and Techniques*. Wiley (1998)
12. Lauria, E.J., Duchessi, P.J.: A Bayesian Belief Network for IT implementation decision support. *Decision Support Systems* 42(3), 1573–1588 (2006)
13. Loucopoulos, P., Karakostas, V.: *System Requirements Engineering*. McGraw-Hill, Inc., New York (1995)
14. de Melo, A.C., Sanchez, A.J.: Software maintenance project delays prediction using Bayesian Networks. *Expert Systems with Applications* 34(2), 908–919 (2008)
15. Meziane, F., Vadera, S. (eds.): *Artificial intelligence applications for improved software engineering development: new prospects*. IGI Global, Hershey (2010)
16. Orellana, F.J., Cañadas, J., del Águila, I.M., Túnez, S.: INSCO requisite - a Web-Based RM-Tool to support hybrid software development. In: *International Conference of Enterprise Information System ICEIS*, Barcelona, Spain, vol. (3-1), pp. 326–329 (2008)
17. Pearl, J.: *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufman, San Mateo (1988)
18. Pendharkar, P., Pendharkar, P., Subramanian, G., Rodger, J.: A probabilistic model for predicting software development effort. *IEEE Transactions on Software Engineering* 31(7), 615–624 (2005)
19. Radlinski, L., Fenton, N., Neil, M.: Improved Decision-Making for Software Managers Using Bayesian Networks. In: *11th IASTED Int. Conf. Software Engineering and Applications (SEA)*, pp. 13–19. Acta Press, Cambridge (2007)
20. del Sagrado, J., del Águila, I.M.: A Bayesian Network for Predicting the Need for a Requirements Review. In: Meziane, F., Vadera, S. (eds.) *Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects*, pp. 106–128. IGI Global, Hershey (2010)
21. del Sagrado, J., del Águila, I.M., Orellana, F.J.: Requirement selection: Knowledge based optimization techniques for solving the next release problem. In: *6th Workshop on Knowledge Engineering and Software Engineering (KESE 2010)*, pp. 40–51. CEUR-WS, Karlsruhe (2010)
22. Sommerville, I.: *Software Engineering*. Addison-Wesley Longman Publishing Co., Inc., Boston (2006)

Dynamic Bayesian Network Factors from Possible Conflicts for Continuous System Diagnosis

Carlos J. Alonso-Gonzalez¹, Noemi Moya¹, and Gautam Biswas²

¹ Department of Computer Science University of Valladolid, Valladolid, 47011, Spain
{calonso, noemi}@infor.uva.es

² Institute for Software Integrated Systems, Nashville, TN 37235, USA
gautam.biswas@vanderbilt.edu

Abstract. This paper introduces a factoring method for Dynamic Bayesian Networks (DBNs) based on Possible Conflicts (PCs), which aim to reduce the computational burden of Particle Filter inference. Assuming single fault hypothesis and known fault modes, the method allows performing consistency based fault detection, isolation and identification of continuous dynamic systems, with the unifying formalism of DBNs. The three tank system benchmark has been used to illustrate the approach. Two fault scenarios are discussed and a comparison of the behaviors of a DBN of the complete system with the DBN factors is also included. Comparison has confirmed that DBN computation is more efficient for factors than for the complete DBN.

Keywords: Dynamic Bayesian Networks, Possible Conflicts, Model-based Diagnosis, Fault Identification, Fault Detection and Isolation.

1 Introduction

The increasing complexity of current engineering systems and the increasing demand on their safe and reliable operation even in the presence of system faults, makes fault diagnosis an essential tool. Due to the complexity of these systems, formal methods are required for systematic design, analysis, and implementation of system diagnosers. Model-based diagnosis provides a formal framework to achieve these objectives.

Among the different approaches to model based diagnosis, stochastic approaches are mandatory when we face uncertainty both on the model parameters and the sensors, particularly in noisy environments [8]. Among stochastic approaches, Dynamic Bayesian Networks (DBNs) [7] play an important role.

DBNs have been applied [25] to fault diagnosis because they allow estimating state variables of a dynamic system without the usual Gaussian assumption for noise and modeling errors, which no longer apply when faults occur [1]. Its major drawbacks are computational complexity of learning and inference procedures. In model based diagnosis, network structure and coefficients may be obtained from models. Real time inference has been tackled with Particle Filtering [1].

A problem with Particle Filtering is 'sample impoverishment': less weighted samples tend to disappear. Importance sampling may reduce this effect that is especially harmful for diagnosis: faulty states have small probabilities. [11] proposes solving this

problem using multiple DBNs: a nominal DBN to track the system in normal operation and, under single fault hypothesis, a DBN to model each fault. The fault hypotheses are tracked in parallel by their associated faulty DBN. Eventually, the DBN which best fits observations provides fault isolation and fault identification. The major drawback of this proposal is the computational complexity of hypotheses tracking, because each DBN models the whole system plus the hypothesized fault.

Factoring DBNs may reduce the computational complexity of inference [11]. In this paper, a factoring approach based on Possible Conflicts (PCs) [9] is proposed, consisting of first decomposing the system with PCs and afterwards obtaining the DBN factors from the PCs. This approach has two advantages: structural observability of the factors is warranted [6] and fault detection and isolation may be performed on the standard framework of Consistency Based Diagnosis [10] with nominal DBN factors. Fault identification in a predictive approach requires considering fault modes [4]. Consequently, the DBN factors are modified to model the fault and tracking the faulty system.

Along the paper, the three tank system benchmark is used to illustrate several concepts. Section 2 provides a basic background about DBNs. Section 3 discusses a method to derive DBNs from PCs. Section 4 introduces the diagnosis architecture to perform fault detection, isolation and identification with PC factored DBNs. In Section 5 two fault scenarios are examined. Section 6 compares the performance of the DBN of the complete system with the DBNs factors. After Discussion, Conclusions are stated.

2 Dynamic Bayesian Networks Background

Dynamic Bayesian Networks are a probabilistic temporal model representation of a dynamic system. Basically, a DBN is a two slices Bayes Network (BN). Assuming that the system is time invariant and a First Order Markov process, two static and identical BN connected by inter slice arcs are enough to model the system [7]. Inter slices arcs model system dynamics. Intra slice arcs model instantaneous (algebraic) relations.

The system variables (X, Z, U, Y) represented in a DBN are the inputs (U), the state variables (X), the observed or measured variables (Y) and, in some cases, other hidden variables (Z). Once we have the nodes, we need to define the parameters of the model, which are the state transition model (graphically represented by the inter slice arcs) and the sensor model (represented by intra slice arcs).

Exact inference in DBNs is not computationally tractable. Hence, Monte Carlo simulation methods are used for approximate inference, particularly Particle Filter algorithm [5]. The unknown continuous stochastic distribution of the state is approximated by a discrete distribution obtained by weighted samples. After propagation of the state, the weights are updated with current observations. In this work, we assume a Gaussian distribution.

Figure 1 shows the three tank system in a) and its DBN model in b). There are three available measurements: (1) the flow out of tank 1 (F_1), (2) the flow between tank 1 and tank 2 (F_{12}) and (3) the flow out of tank 3 (F_3). They are represented in the network by f_4 , f_6 and f_{16} respectively. F_{in} is a constant input, represented by node f_1 . Nodes e_2 , e_8 and e_{14} are the state variables, the pressures at the bottom of each tank. Hence, $X = \{e_2, e_8, e_{14}\}$, $U = \{f_1\}$, $Y = \{f_4, f_6, f_{16}\}$ and $Z = \{\}$.

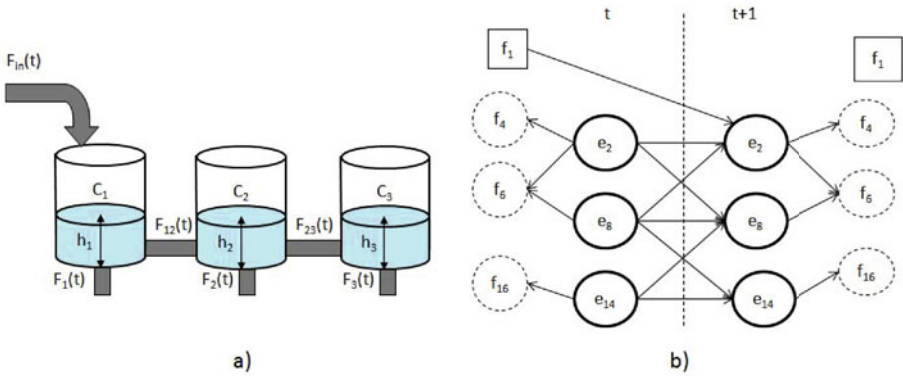


Fig. 1. a) Three tanks system. b) DBN modeling the three tanks system in a).

The DBN structure and parameters may be obtained from the state space equations of the system.

3 Obtaining DBN Factors from PCs

3.1 Possible Conflicts Background

Possible Conflicts (PCs) is a compilation technique for consistency based diagnosis of dynamic systems [9]. Essentially, PCs are minimal over determined subsystems with analytical redundancy.

Possible conflicts can be automatically derived from a hypergraph model of a system. This hypergraph is just an abstract representation of the system equations in state space form. Hyperarcs of the hypergraph represent an equation (more generally, a constraint) and the nodes included in the hyperarc are the variables of the equation (i.e. constraint).

Each PC has associated a directed hypergraph called *Minimal Evaluable Model (MEM)*. Nodes of the directed hypergraph represent variables of the system and directed hyperarcs represent a constraint with a causal assignment. From a given MEM, a computational model of a PC can be directly obtained replacing each hyperarc by its corresponding equation. A distinguished node in a MEM is the discrepancy node, that is the node where redundancy manifests.

The three tank system of Figure 1 a) has three PCs. Figure 2 a) shows the MEM of PC1 for this system. $ec1_1$ models the mass balance at tank 1, $ec4_1$ models the flow out of tank 1 and $ec12_1$, $ec13_1$ and $ec14_1$ model the sensors. The dash arc is a differential constraint in integral causality. The discrepancy node in PC1 is f_4 .

A relevant aspect of PCs is that each possible conflict identifies a subsystem that is independent, in the sense that it can be analyzed in isolation, because PCs are structurally observable [6]. Moreover, they are minimally redundant. These properties make them an interesting tool to decompose DBNs.

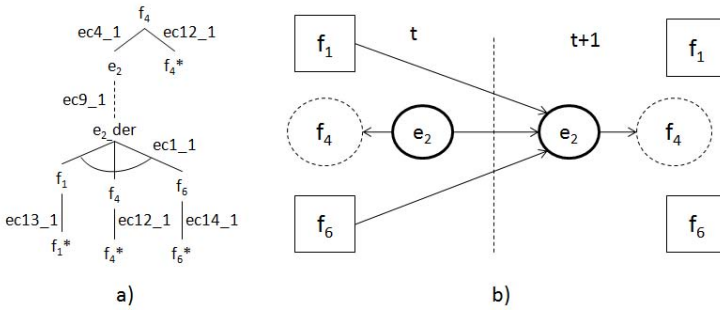


Fig. 2. a) Possible Conflict 1 of the three tanks system. b) DBN model for Possible Conflict 1.

3.2 Generating a DBN from a PC

In this work we propose a method to obtain a DBN from the associated *MEM* of a PC.

Proposition 1. *Those Possible Conflicts with a MEM containing:*

- **Condition 1**, a state variable and a differential arc,
- **Condition 2**, a path made only of instantaneous arcs from an estimated state variable to a discrepancy node that is observed,

provide the minimal structural description of a DBN for the subsystem defined by the possible conflict.

Condition 1 is required to have a dynamic system. Condition 2 is necessary to avoid an empty sensor model in the DBN.

The demonstration of **Prop. 1** is constructive and generally requires two steps:

- **Structure:** Generate DBN structure from nodes and hyperarcs of the related *MEM* according to the criteria of Table 1.
- **Simplification:** For any state variable which is conditionally dependent only on input nodes, replace that state variable and inputs by a new input node, according to algebraic *MEM* computation.

The *Structure* step defines the initial sets of nodes and arcs of the DBN. Second step of the construction process just simplifies the DBN, eliminating state variables that are algebraically estimated from known inputs and observed variables in the original *MEM*.

Figure 2 b) shows the DBN obtained from the PC1 of the three tank systems, applying just the *Structure* step. None of the PCs of this system needs to perform the *Simplification* step to generate the DBN.

4 Diagnosis Architecture with DBNs and PCs

Factored DBNs from Possible Conflicts allows tackling all the stages of model based diagnosis, that is, fault detection, fault isolation and fault identification, in the Consistency Based Diagnosis framework with fault models in a predictive approach. Figure 3 shows the architecture of the system.

Table 1. How to derive the DBN structure from a PC hypergraph: On the left there are the equivalence between nodes in the hypergraph of a MEM and nodes in the DBN. On the right, the equivalence between relations in the hypergraph of a MEM and arcs in the DBN.

PCs	DBNs	PCs	DBNs
Inputs (U)	Inputs	Differential constraint	Inter slice arc for related state variable
Observation of the discrepancy node	Observation (sensor model)	Path from a state variable to a state variable, including only one differential constraint	Inter slice arcs from state variable to state variable
Any other observation	Input	Path from an observation or input to a state variable, including only one differential constraint and no additional state variables	Inter slice arcs from nodes to state variable
States	States	Paths without differential constraints, starting or ending at a state variable	Intra slice arcs

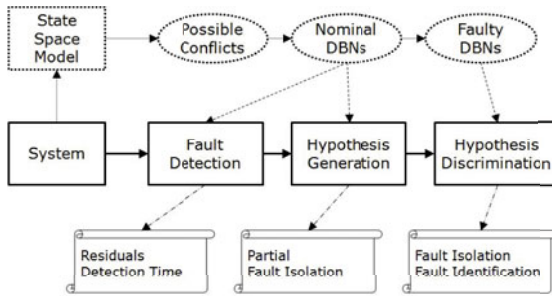


Fig. 3. The diagnosis architecture integrating DBNs and PCs

4.1 Fault Detection

Nominal DBN factors are obtained off line from the system model through PCs decomposition. The three resultant DBN factors for the three tank system are shown in Figure 2 b) and Figure 4. These DBN factors are run in parallel to perform fault detection. A ztest [3] on the residual of tracked variables is used to decide on detection of each DBN factor.

4.2 Fault Isolation

In a predictive approach, fault isolation requires introducing fault modes. We have opted for a simple abrupt fault model [11].

Abrupt Fault. An abrupt fault is characterized by a fast change in a parameter value. The temporal profile of a parameter with an abrupt fault, $p^a(t)$ is given by:

$$p^a(t) = \begin{cases} p(t) & t < t_f \\ p(t) + b(t) = p(t) + \sigma_p^a & t \geq t_f \end{cases}$$

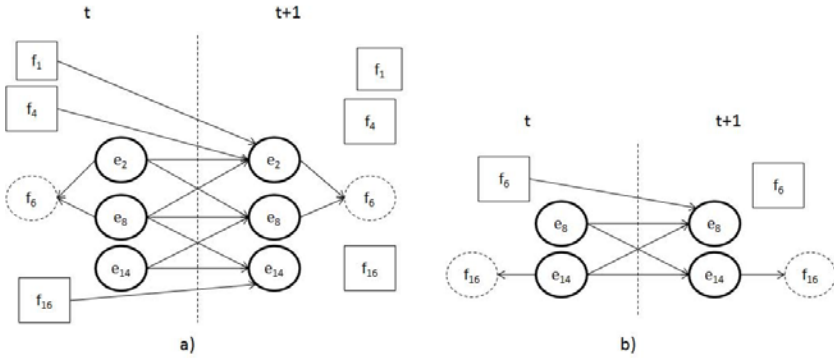


Fig. 4. DBN model for a) the PC2 and b) the PC3 of the three tank system

where σ_p^a models the absolute change of the parameter value. For all the considered faults, σ_p^a is set to 10% of the nominal value of the parameter.

There are 8 possible abrupt faults: in the capacitances of each tank (C1, C2 and C3), in the resistance of the output of each tank (Rv1, Rv2 and Rv3) and in the resistance of the flow between tanks (Rv12 and Rv23). The fault signature matrix of the DBN factors is the same as the fault signature matrix of the PCs of the system (see Table 2). We use this fault signature matrix to generate Reiter candidates, which are updated if new observations generate new detections, assuming non intermittent faults.

Table 2. Transposed fault signature matrix of the three tank system

	C1	C2	C3	Rv1	Rv12	Rv2	Rv23	Rv3
PC1	1			1				
PC2	1	1	1		1	1	1	1
PC3		1	1			1	1	1

For complexity reasons, we limit fault identification to single faults. DBN factors for each fault mode are obtained from DBN factors of the nominal system according to [11] proposal. Nominal DBN factors are extended with an additional node for the faulty parameter. If some network node is conditionally dependent on the new node, an edge is added from the new node to the 'not conditionally independent' node. Figure 5 shows the faulty network factor obtained from PC1 for an abrupt fault in the capacitance of tank 1. For each DBN factor it is necessary to build as many faulty DBNs as indicated in the fault signature matrix.

Fault isolation requires tracking the system with faulty DBNs. For each single fault candidate a faulty DBN factor has to track the system. If a new detection allows reducing the number of single fault candidates, the corresponding fault hypotheses are rejected and the associated DBN factors no longer track the system. Eventually, one of the faulty DBNs will converge identifying the new value of the parameter.

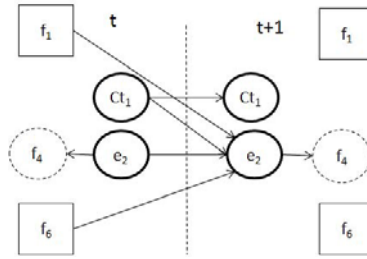


Fig. 5. DBN of PC 1 of the three tanks system with a fault in the capacitance of tank 1

5 Fault Scenarios

We have developed two fault scenarios for the three tank system: (1) an abrupt fault in the capacitance of tank 1 and (2) an abrupt fault in the resistance out of tank 3. In both cases the fault magnitude is 10% of the nominal value of the parameter. We have simulated for 10,000 time steps, starting with the three tanks empty and injecting the faults at time stamp 2,000. Simulink has been used to generate data of the faulty system. A 5% and a 0.5% Gaussian noise has been added to sensors and input, respectively. The number of particles used in the Particle Filter algorithm has been 500. Ztest has been applied to decide on network detection and also on network convergence for fault identification.

5.1 Abrupt Fault in C1

Fault detection is performed with the nominal DBN factors (see Figure 6 I). The DBN from PC1 (PC2) detect the fault at time 2,001 (2,002) (Figure 6 I): a, b) According to the fault signature matrix of the system, Table 2 the factor from PC3 does not detect the fault (Figure 6 I): c).

Hence, from time 2,002 there is only one single fault candidate: C1. We have run the faulty DBN of PC1 for a fault in C1 (Figure 5) starting 50 time steps before the fault is injected, to launch simulation from a known system state with nominal behavior. The behavior of the network is shown in Figure 6 II). Convergence time is 389 time steps, (339 after fault injection).

5.2 Abrupt Fault in Rv3

Like in the previous scenario, fault detection is performed with the three nominal DBN factors. Now, only the factor from PC3 detects the fault, at time 2,007.

In this case, we have 5 single fault candidates (C2, C3, Rv2, Rv23 and Rv3) and we have to run the faulty DBNs from PC3 for all these faults to check which one converges.

The DBN from PC3 with the extra node for the fault in Rv3 is able to track the state variables and it also gives us a good estimation of the parameter after the fault. Table 3 presents the results for both experiments. Although not displayed, faulty DBNs modelling a different fault do not converge.

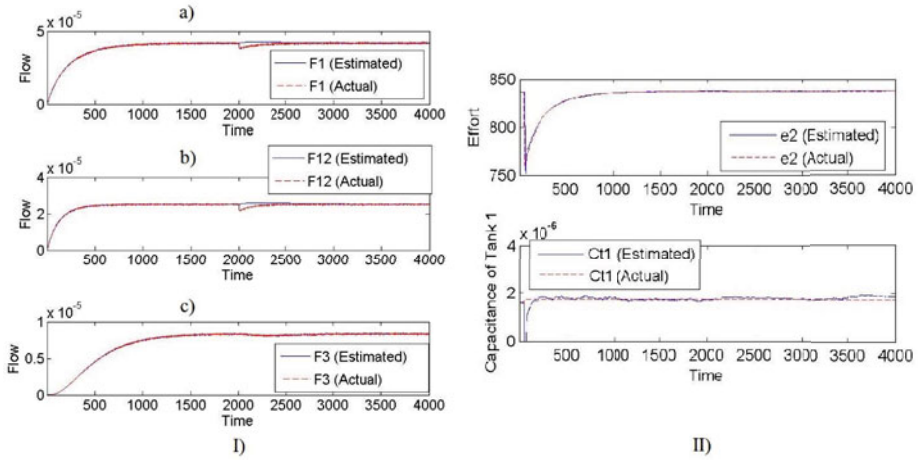


Fig. 6. I) Observed variables tracked with nominal DBNs of a) PC1, b) PC2 and c) PC3 for an abrupt fault in the capacitance of tank 1. II) State variable tracked with the faulty DBN of PC1 for the same fault and the estimation of the parameter C1.

6 Comparing Complete DBN and DBN Factors Performance

This section presents a quantitative comparison of the performance of the DBN of the complete system and the performance of the DBN factors. We have estimated mean execution, detection and convergence times. Parameter value convergence is also considered. All experiments have been repeated ten times. Table 3 sum up the results mentioned before. In the *Nominal DBNs* columns, the column *Execution Time*, shows execution time for 10.000 time steps. As it was to be expected, DBN factors require less computation time than the original DBN. Faults are injected at time 2000. Columns *Detc. C1A* and *Detc. Rv3A* show that detection time is similar for every network considered. There are no false positive detections.

Table 3. Performance summary for the two faulty scenarios with nominal and faulty DBNs

	Nominal DBNs			Faulty DBNs					
	Exec. time	Detc. C1A	Detc. Rc3A	Exec. C1A	Exec. Rv3A	Conv. C1A	Conv. Rv3A	MSE C1A	MSE Rv3A
Complete	84.25	2, 001	2, 008	46.66	67.15	$1.15 \cdot 10^{-3}$	$3.78 \cdot 10^{-2}$	$2.62 \cdot 10^{-8}$	$2.77 \cdot 10^{-4}$
PC1	71.14	2, 002		43.43		$3.89 \cdot 10^{-2}$		$2.06 \cdot 10^{-8}$	
PC2	77.77	2, 001		45.4		$4.95 \cdot 10^{-2}$		$1.67 \cdot 10^{-8}$	
PC3	78.4		2007		63.31		$3.78 \cdot 10^{-2}$		$2.81 \cdot 10^{-4}$

For fault identification, simulation starts 50 time steps before the fault is injected and simulation time extends to 8.050 seconds. Table 3 under *Faulty DBNs*, shows execution time (first two columns) for faulty networks, that are also smaller for DBN factors. Third and fourth columns show convergence time for each fault. Compared with the complete

DBN, convergence time is smaller for the fault in capacitance of tank 1, and it is equal for the fault in resistance Rv3.

Finally, in the same table, the last two columns have the Mean Square Error (MSE) of the estimated parameter. Error is smaller for PC1 and PC2 for faults in capacitance of tank 1, but it is slightly bigger for PC3 for fault of resistance Rv3. The variance of convergence times and MSEs, not displayed in the table, are smaller for the DNBs factors than for the complete network by one and two, respectively, orders of magnitude.

7 Discussion

The two faulty scenarios considered in this paper show that PCs decomposition from a state space representation of a system allows using a unique formalism, DBNs, to perform fault detection, isolation and identification with a simple architecture.

Interpretation of quantitative efficiency measures should be done carefully, because final quantitative figures depend on several factors, including fine parameter tuning of the diagnoser. All experiments have been performed with Matlab, with the same Particle Filtering software and on the same machine. Ztest parameters have been selected conservatively, favoring the convergence to the real parameter value against fast fault identification. These initial experimental results indicate that the approach is computationally advantageous even for a small system like the three tanks system. Further research is needed to obtain confident conclusions about parameter value estimation accuracy. The DBN factor from PC2, which keeps all the state variables of the system and a single observation, provides the best estimation of the new value of the capacity in tank 1. However the DBN factor from PC3, with a simpler structure, estimates the value of Rv3 slightly worse than the DBN of the complete system.

A related approach to fault detection, isolation and identification of continuous systems with factored DBNs is presented in [11]. Their proposal to obtain DBN factors is based on conditional independence. They define DBN factors as a subset of random variables of the complete DBN, conditionally independent of the variables in all other DBN factors, for a given set of observations. Afterwards, they iteratively have to merge factors until an observable DBN factor is obtained.

Both approaches have some similarities, like eliminating state variables that can be computed by algebraic relations and assuring that the resulting factors are observable. However, the factoring methodology is different. In [11] network splitting does not consider observability, which has to be recovered later merging unobservable factors with other factors. In contrast, PCs decomposition warrants the observability of the factors, which also assures their conditionally independence. Factoring is more systematic with the PCs decomposition. Deriving factors from the PCs has the advantage that all minimal factors with analytical redundancy are found. Minimal factor are desirable because they have the potential to maximally reduce computing time on a simulation based approach, particularly for fault identification. Nevertheless, further research is needed to characterize both approaches on those dimensions and to compare their performance on complex, real systems.

8 Conclusions

This work has presented a method to factor Dynamic Bayesian Networks: generate the factors using Possible Conflicts. These factors are minimal redundant structurally observable subsystems. Factoring is desirable because it reduces system complexity, simplifying its analysis and enabling the design of more efficient diagnosers. Structural observability of the factors is needed not only to compute the state variables of the factors, but also to assure their conditional independence of the other minimal factors.

Based on DBN factors, a unified solution has been proposed to Consistency Based fault detection, isolation and identification. Two scenarios have been developed on the three tanks system benchmark to illustrate the proposal. A quantitative comparison of the performance of the DBN of the complete system and the DBNs factors has also been done, in terms of execution, detection, and convergence time plus parameter estimation error. Comparison has confirmed that DBN computation is more efficient for factors than for complete DBNs.

Acknowledgments. This work has been partially supported by the Spanish Office of Science and Innovation (MCINN) through grant TIN2009-11326.

References

1. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing* 50(2), 174–188 (2002)
2. Dearden, R., Clancy, D.: Particle filters for real-time fault detection in planetary rovers. In: *Proceeding of the 12th International Workshop on Principles of Diagnosis*, pp. 1–6 (2001)
3. Gelso, E.R., Biswas, G., Castillo, S.M., Armengol, J.: A comparison of two methods for fault detection: a statistical decision, and an interval-based approach. In: *Proceeding of the 19th International Workshop on Principles of Diagnosis, DX 2008* (2008)
4. de Kleer, J., Williams, B.: Diagnosing with behavioral modes. In: *Eleventh International Joint Conference on Artificial Intelligence, IJCAI 1989* (1989)
5. Koller, D., Lerner, U.: *Sampling in factored dynamic systems*. In: *Sequential Monte Carlos Methods in Practice*. Springer, Heidelberg (2001)
6. Moya, N., Biswas, G., Alonso-González, C., Koutsoukos, X.: Structural observability. application to decompose a system with possible conflicts. In: *Proceeding of the 21th International Workshop on Principles of Diagnosis, DX 2010* (October 2010)
7. Murphy, K.P.: *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. thesis, University of California, Berkeley (2002)
8. Narasimhan, S.: Automated diagnosis of physical systems. In: *Proceedings of ICALEPCS 2007*, pp. 701–705 (2007)
9. Pulido, B., Alonso-Gonzalez, C.: Possible conflicts: a compilation technique for consistency-based diagnosis. Part B: Cybernetics, *IEEE Transactions on Systems, Man, and Cybernetics* 34(5), 2192–2206 (2004)
10. Reiter, R.: A theory of diagnosis from first principles. *Artificial Intelligence* 32, 57–95 (1987)
11. Roychoudhury, I., Biswas, G., Koutsoukos, X.: Designing distributed diagnosers for complex continuous systems. *IEEE Transactions on Automation Science and Engineering* (2009)

Planning and Execution in a Personalised E-Learning Setting

Lluvia Morales¹, Antonio Garrido², and Ivan Serina³

¹ University of Granada, Spain

² Universitat Politècnica de València, Spain

³ Free University of Bozen-Bolzano, Italy

lluviamorales@decsai.ugr.es, agarridot@dsic.upv.es, ivan.serina@unibz.it

Abstract. The main aim of e-learning is to provide a learning route where activities are tailored to individual necessities. But this is not always enough, as this route needs to be executed in a real learning management system where some discrepancies (between the real and expected situation) may appear. In this paper we focus on the generation of these routes from a planning perspective, but also on the monitoring and execution of the routes and, in case of significant discrepancies, provide a planning approach for adapting the route —rather than generating a new one from scratch. We demonstrate that this approach is very valuable to maximise the *stability* of the learning process, and also for the performance and quality of the learning routes.

Keywords: applications of AI, e-learning, planning, personalisation of e-learning routes.

1 Introduction

E-learning is, in essence, a multidisciplinary field that takes advantage of the current advances in technology and integrates many techniques from different fields, such as educational theories, profile identification and modelling, knowledge representation, AI methods and optimisation procedures among others.

The minimal component of e-learning is a Learning Object (LO), which is an interoperable resource to be used in flexible learning routes that support and enhance learning. Thus, LOs have been likened to LEGO bricks and the way they can be stacked to form bigger structures and reused once and again. In other words, the utmost LO reusability cannot be achieved by considering the LOs as isolated components, but as aggregated elements for large courses to be eventually executed by students. From this execution perspective, we have to deal with two challenging issues. First, how to build the right sequence of LOs for each student, i.e. to provide a learning route where LOs and activities are tailored to the specific needs, objectives, background and, in general, profile of each student (personalised learning [5,10,14]). Second, how to monitor the execution of the learning route, check its progress and act when discrepancies (differences w.r.t. the expected state) appear, i.e. to provide a flexible adaptation

process that does not ignore the original student's interests and tries to reuse the original route as much as possible. This paper mainly builds on these two issues and contributes with an AI planning approach to: i) model and encode learning courses and students' profiles as planning problems; ii) solve these problems to find plans, i.e. learning routes, that entirely fit the students' interests; iii) monitor the execution of the learning routes checking its validity; and iv) adapt the route in the event of a discrepancy that prevents the execution of the route.

The structure of the paper is as follows. Section 2 presents some related work for course composition and personalisation of learning routes. In section 3 we propose our schema for planning e-learning routes, give a short description on the knowledge representation stage to compile the corresponding planning problem and detail the importance of stability in e-learning. How we can interleave execution, monitoring and adaptation of learning routes is deeply explained in section 4. Section 5 shows our experimental results and, finally, section 6 concludes the paper.

2 Related Work

Course composition has been traditionally seen from two perspectives: i) adaptive courseware generation, and ii) dynamic courseware generation. In the former, the idea is to sequence an individualised course taking into account specific learning goals and the student's previous knowledge. Thus, the main goal is to ensure that a student completes all the activities that an instructor deems important, which makes the objective instructor-centered [1]. Many techniques have been successfully applied to generate personalised courses as a means to bring the right content to the right person, such as adjacency matrices, integer programming models, neural networks and AI planning [3,5,7,8]. In the latter, the system observes the student's progress during his/her interaction with a general course and dynamically adapts it according to the specific student's needs and requirements [11,13]. Hence, the goal is to assist students in navigating in a complex information space in order to achieve whatever goal they choose, making this type of hypermedia technique student-centered. In other words, the adaptive generation selects LOs from a given repository in a way which is appropriate for the targeted individuals, whereas the dynamic generation provides a more accurate browsing associated with an on-line course in an optimal order, where the optimisation criterion takes into consideration the student's background and performance.

There are a few works, such as [2,14], that combine the two previous perspectives as part of their own intelligent tutoring systems. But in most cases, once the route is created, the monitoring part to check its execution is missing. This represents an important limitation, because it is not only important to generate a personalised route, but also to check how it is navigated and executed by the student, and adapted if some discrepancies (between the real and expected situation) appear. Revisiting the LEGO metaphor, it does not suffice with having the plan of a big structure because we also need to put it into practice. And if

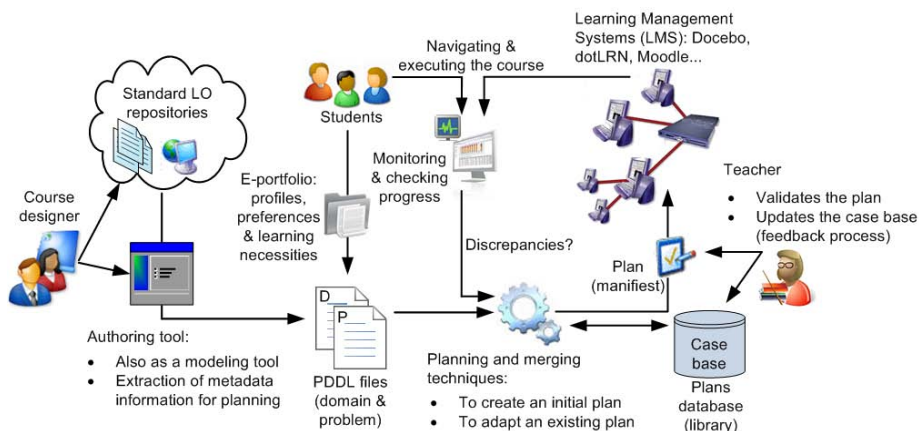


Fig. 1. Overall structure for using planning in an e-learning setting

one brick is missing when creating the structure, we do not discard the whole structure but try to replace the missing brick with others that play a similar role (and reuse as much part of the original structure). This paper overcomes this limitation by using e-learning and planning standards, making it more general and applicable.

3 A General Approach for Planning E-Learning Routes

3.1 Description of the Structure at a Glance

Fig. 1 depicts the general structure of our approach. The idea is to define a course by using LOs available in (Web) repositories. Once the course has been defined and the students' profiles—in terms of background, learning styles and interests—have been modelled, an automatic translator compiles all this information as a planning problem to be solved by a standard planner. The planner generates a plan, i.e. a learning route, that is validated by one teacher (and also stored in a plan library). This route is uploaded to a Learning Management System (LMS) as a plan manifest that allows a student to navigate through his/her tailored route. The LMS also monitors the execution of each route and if any discrepancy is found between the real and expected state, a new planning iteration is launched to fix (adapt/replan), or improve, the learning route. Note that the new route should not significantly differ from the original one, so keeping a high stability record is important in e-learning.

3.2 Compilation of Domain+Planning Problem

This is a knowledge representation stage and provides the foundations for using planning technology. It consists in mapping the information about the course

(the LOs, their relationships, technical and educational requirements) into PDDL actions, which define the planning domain. LOs are usually labelled by an XML metadata format, such as LOM [9]. Thus, the planning domain is generated by iterating all over the LOs of the course to generate one PDDL action per LO. This compilation is very efficient (polynomial time), as each action comprises four entries automatically extracted from the values of the LO metadata specification: i) name of the LO; ii) duration of the LO (learning time); iii) pre-conditions, based on the profile's dependence plus the relations defined in its metadata; and iv) effects, based on the learning outcomes. For further details about this compilation see [5].

On the other hand, the planning problem is compiled by extracting the relevant students characteristics from his/her e-portfolio, which are obtained from the XML files in IMS-LIP and IMS-QTI standards. In addition to the initial state (background) and learning goals per student, it can also include the metric to be optimised, such as finding the shortest learning route or the one that maximises a given reward, score or learning utility.

The generation of a PDDL planning problem facilitates the use of independent solvers, which provides a nice approach to abstract the e-learning specific features from the planning details. When a plan is found, as a sequence of LOs that best suits the student, it is displayed in a LMS (see Fig. 11).

3.3 The Importance of Plan Stability

The ultimate objective in planning is to construct plans for execution. However, when a plan is executed in a real environment it can encounter differences between the expected and actual context of execution. These differences can manifest as divergences between the expected and observed states of the world, or as a change in the goals to be achieved by the plan. In both cases, the original plan must be replaced with a new one. In replacing the plan an important consideration is *plan stability*. As proposed in [4], we use this term to refer to a measure of the difference a process induces between an original (source) plan and a new (target) plan. In general, we will be considering cases where the new plan is intended to solve a different, although related, problem to the one solved by the original plan. This means that there will inevitably be a difference between the plans. In the e-learning context it is extremely important to preserve as much as possible the LOs planned for each single student; in fact, it could be extremely disappointing if a completely new sequence of LOs is proposed to the students given a change in the current state, and it should be avoided as much as possible, especially if the original LOs can still be used.

Then, we decided to use a simple but very effective notion of plan stability [4,12] based on the distance in terms of number of different LOs between two learning plans. Following the formalization proposed in [4], the *distance* between two plans is simply defined as the number of actions (LOs in our context) that appear in the first plan and not in the second, plus the number of actions that appears in the second plan and not in the first one. Given an initial learning plan that is no longer valid due to a change of the current state or to a change of the

domain representation, the notion of *plan stability* is simply defined in terms of the *distance* of the new solution plan w.r.t. the original plan.

4 Executing, Monitoring and Adapting E-Learning Routes

The use of LMSs is important, mainly for the students but also for the teachers. The LMS identifies the instructional design per student, i.e. his/her learning route, can be visualised under the IMS-CP or SCORM specifications following the compilation criteria described in section 3.2. The LMS is therefore useful not only for navigation matters, but also to automatically monitor the student's progress and detect significant discrepancies between the current situation and the scheduled (expected) situation in a kind of *checkpoint* (see Fig. 2). These discrepancies appear due to changes on the background/profile information, the temporal constraints, the resource availability, or the execution of the LOs in themselves. Some examples of this are:

- The student's background is externally changed. For instance, the student is involved in an external language course, or has worked with many LOs in that language, and consequently (s)he has become more proficient in such a language. These improvements in the student's skills will allow him/her to choose now from a higher number of LOs.
- The learning style orientation of the student changes throughout the course execution, which entails a revision of the remaining LOs of the course. Some of them will remain valid, but others should be replaced to fit the new student's profile.
- The student has extra temporal constraints (getting a new job or being sick), and now (s)he has less time to accomplish the goals of the course, thus being unable to perform some LOs. This is likely to create an inconsistency when using *tightly-agenda* LOs.
- There is a change in the availability of the equipment, which is temporary unavailable. Or perhaps the student has now a better-equipped computer. In both cases, the learning route may require an adaptation process.
- During the course execution the student might fail a test or questionnaire, that is a checkpoint LO used to evaluate his/her comprehension on the course objectives. If this comprehension shows a low score, the student's performance will not be enough to attain the learning outcomes.

As can be noted, there are both *positive* and *negative* discrepancies. Positive discrepancies, such as having more available resources or when students' abilities are improved, do not invalidate the learning route, but they could lead to a better quality route, i.e. shorter makespan or higher reward plans. On the contrary, negative discrepancies make the learning route no longer executable, e.g. some resources are unavailable or the student fails an evaluation activity.

The changes in students' background, learning styles and temporal constraints must be modified directly by the students using the LMS interface. Changes related to the resource availability are usually updated by teachers, and scores

of evaluation activities can be input by the teachers or automatically calculated by the LMSs. With all these changes, a new (planning) problem with the same learning goals —although they can be also changed if desired— and a new initial state is created. After this, our way of proceeding is depicted in Fig. 2. When changes in the student’s profile are detected, we simulate the execution of the remaining part of the learning route, starting from the new state, in order to identify if it contains flaws, i.e. whether the prerequisites of LOs and the goals are satisfied or not (this validation process can be computed efficiently in polynomial time w.r.t. the number of actions in the remaining part of the plan [6]). If an inconsistency is detected, it is highlighted to the teacher and (s)he can decide whether to repair it manually or to ask for a new plan to the planner that will fix the flaw, that is automatic adaptation. If no inconsistency is detected, a new schedule of the remaining LOs is provided to the student in order to better satisfy his/her requirements and time availabilities; note that this new schedule can be simply computed in polynomial time w.r.t. the number of LOs and resources involved, and does not require any kind of validation by the teacher since the LOs have not changed. Moreover, the student and the teacher can also ask the planner if a new plan of better quality, according to the new student’s profile and the current resources, can be found. Anyway, once a new plan is computed by our system it must be always validated by the teacher before its execution, and the plan stability, in terms of number of actions of the original plan, is of capital importance to reduce the teacher’s overhead. When the plan execution finishes and all the students’ goals are satisfied, the corresponding plan is stored in the case base (plan library), if not already present, closing in this way the learning cycle as shown in Fig. 11.

5 Experimental Results

In this section we test the effectiveness of our adaptation approach *vs.* plan generation techniques when discrepancies appear while executing the learning routes. Particularly, we focus on: i) the CPU time required to repair (adapt or replan) the routes, ii) the best qualities in terms of higher reward plans, and iii) the best stability that can be obtained in a given deadline. We use 2 real, different Moodle courses (planning domains), on Discrete Maths and Natural Sciences, which are medium- and large-size, respectively. For each of the 2 courses, we have created 4 initial configurations (with 20, 40, 60 and 80 different students, respectively), and defined 10 variants per configuration, thus considering 88 planning problems in total (the 80 variants plus the 8 initial configurations). Each variant artificially simulates the changes that may occur during the route execution in an incremental way. That is, in the first variant some equipment is no longer available. The second variant maintains these changes and includes restrictions on the students’ availability; and so on for the other variants.

In addition to our adaptation approach, implemented on LPG-ADAPT [4], we have used two state of the art planners, SGPLAN6 and LPG [1]. Since LPG and

¹ For a further description of these planners see <http://ipc.icaps-conference.org>

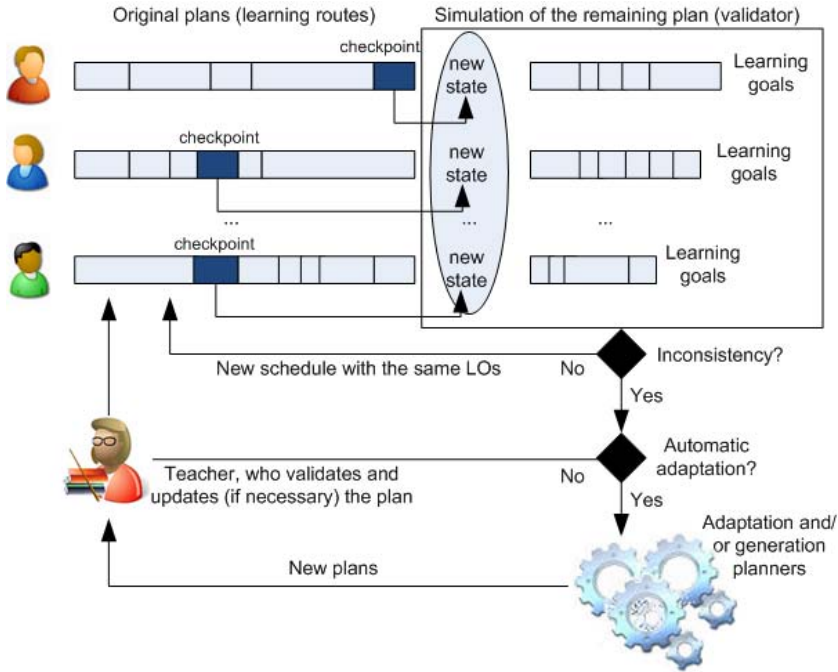


Fig. 2. Schema for monitoring and adapting the e-learning route

LPG-ADAPT are stochastic incremental planners and different executions usually differ, we have performed 5 runs for each planning problem and taken the median of these values for our plots. All tests were performed on an Intel(R) Xeon(TM), CPU 2.40GHz, 2GB of RAM, and censored after 10 minutes. In our tests, the input plan (i.e. the learning route) to be adapted by LPG-ADAPT was obtained by using the best quality plan generated by LPG and SGPLAN6 on the initial-configuration planning problem used to create the corresponding variants.

Fig. 3 depicts the results: the time taken to produce a solution—the first one for LPG and LPG-ADAPT— (top); the quality of the generated routes (middle); and the stability, in terms of distance of the new routes to the original ones (bottom). We show the best distance and plan quality across all plans produced in the entire optimisation phase². The results demonstrate that plan adaptation is at least as fast as replanning, and usually faster. Obviously, adaptation shows less useful when the changes are significant and fixing the route requires more effort than simply discarding it and rebuilding a new one from scratch. But the benefits for investing this effort can be seen in terms of stability. On the other hand, the adaptation sometimes comes at a price in terms of quality, as the

² Note that the first plan generated by LPG and LPG-ADAPT, the best quality plan and the best distance plan could be different plans. It depends on the teacher's preferences to give more importance to the plan quality or to the plan stability by selecting the most appropriate solution plan during the validation process.

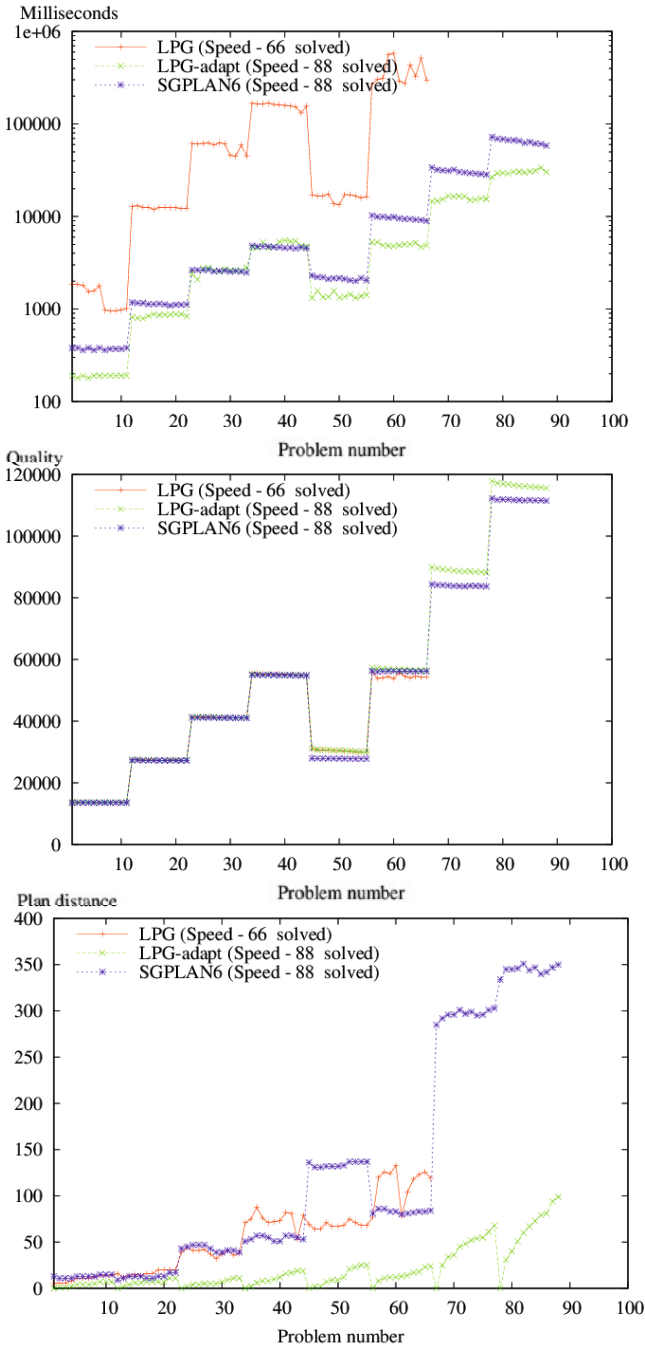


Fig. 3. CPU time (on a logarithmic scale), plan qualities and number of different LOs w.r.t. the input plan of the adaptation —small values are preferable except for quality. We compare our adaptation approach (LPG-ADAPT) *vs.* replanning (LPG and SGPLAN6).

route is adapted to fit a new configuration rather than constructed expressly for it. But our experiments show that the quality for adaptation can be better than for replanning, particularly in the most complex problems (see Fig. 3). We therefore compare the best relative qualities of plans generated by adaptation and replanning within a certain CPU time limit. Finally, the best values for stability are outstandingly achieved in plan adaptation. While replanning generates routes that are consistently very different to the original ones (see SGPLAN6 for a clear example), the differences between the adapted plan and the original plan are very small. This indicator is very appealing in an e-learning setting as the students/teachers do not want to deal with an entirely new learning route after a little change happens during execution. Quite the contrary, students and teachers prefer a kind of *inertia* in the learning routes that facilitates the learning process.

6 Conclusions

Personalisation of e-learning routes is essential for both educational and enterprise organizations, as it supports a continuous and fruitful lifelong learning process. In this paper, we have presented a flexible way that consists in the compilation of the course+students characteristics into a planning problem to find these routes. But once a learning route is generated, how the students execute such a route (and use its LOs) is also a challenging aspect. Monitoring the route may detect inconsistencies that can turn it invalid, and an adaptation process becomes crucial.

We have proposed an approach for executing and monitoring learning routes that uses an adaptation method to repair unexpected discrepancies (and to improve the quality of the original plan when possible). This approach has some advantages, which are the main contributions of this paper: i) it is implemented on top of a standard LMS platform (Moodle), and all the information retrieved and produced is mapped from, and to, e-learning standards; ii) the adaptation technology considers, not only students' preferences on the course, but also teachers'; iii) it allows dynamic changes both on the students' profiles (planning problem) and on the course structure (planning domain); iv) it provides multi-optimisation methods, useful for modern planners and for navigation in LMSs.

As part of our current work, we are addressing two issues. First, to extend the notion of plan stability to deal with extra temporal and resource constraints (e.g. the use of a laboratory at a specific time or the participation of the same set of students in group activities). The idea is to include structural properties of the original plan expressed as *preferences* to be maintained in the new plan. Second, to implement a collection of Web services to be used as a standard interface for LMS-based agents to monitor changes in the student's profile, course composition and description. This will allow us to evaluate our approach in a higher number of real students and situations.

Acknowledgments. This paper was partially funded by the Consolider AT project CSD2007-0022 INGENIO 2010 of the Spanish Ministry of Science and Innovation, the MICINN project TIN2008-06701-C03-01, the Mexican National Council of Science and Technology, the Valencian Prometeo project 2008/051 and the BW5053 research project of the Free University of Bozen-Bolzano.

References

1. Abdullah, N., Davis, H.: Is simple sequencing simple adaptive hypermedia? In: Proc. ACM Conference on Hypertext and Hypermedia, pp. 172–173 (2003)
2. Camacho, D., Pulido, E., Rodriguez-Moreno, M., Carro, R., Ortigosa, A., Bravo, J.: Automatic course redesign: Global vs. individual adaptation. *Journal of Engineering Education* 25(6), 1270–1283 (2009)
3. Castillo, L., Morales, L., Gonzalez-Ferrer, A., Fdez-Olivares, J., Borrajo, D., Onaindia, E.: Automatic generation of temporal planning domains for e-learning. *Journal of Scheduling* 13(4), 347–362 (2010)
4. Fox, M., Gerevini, A., Long, D., Serina, I.: Plan stability: Replanning versus plan repair. In: Proc. 16th Int. Conference on Automated Planning and Scheduling (ICAPS 2006), pp. 212–221. AAAI Press (2006)
5. Garrido, A., Onaindia, E., Morales, L., Castillo, L., Fernandez, S., Borrajo, D.: Modeling e-learning activities in automated planning. In: Proceedings of the 3rd International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS-ICAPS 2009), pp. 18–27 (2009)
6. Howey, R., Long, D., Fox, M.: Validating plans with exogenous events. In: Proc. 23rd UK Planning and Scheduling SIG Workshop, pp. 78–87 (2004)
7. Idris, N., Yusof, N., Saad, P.: Adaptive course sequencing for personalization of learning path using neural network. *Int. J. Advance. Soft Comput. Appl.* 1(1), 49–61 (2009)
8. Kontopoulos, E., Vrakas, D., Kokkoras, F., Bassiliades, N., Vlahavas, I.: An ontology-based planning system for e-course generation. *Expert Systems with Applications* 35(1-2), 398–406 (2008)
9. LOM: Draft standard for learning object metadata. IEEE. rev. February 16, 2005 (2002), http://ltsc.ieee.org/wg12/files/IEEE_1484_12_03_d8_submitted.pdf
10. Peachy, D., McCalla, G.: Using planning techniques in intelligent systems. *International Journal of Man-Machine Studies* 24, 77–98 (1986)
11. Perez-Rodriguez, R., Rodríguez, M., Anido-Rifón, L., Llamas-Nistal, M.: Execution model and authoring middleware enabling dynamic adaptation in educational scenarios scripted with PoEML. *Journal of Universal Computing Science* 16(19), 2821–2840 (2010)
12. Srivastava, B., Nguyen, T., Gerevini, A., Kambhampati, S., Do, M., Serina, I.: Domain independent approaches for finding diverse plans. In: Proc. Int. Joint Conference on AI (IJCAI 2007), pp. 2016–2022 (2007)
13. Ullrich, C., Lu, T., Melis, E.: Just-in-time Adaptivity Through Dynamic Items. In: Houben, G.-J., McCalla, G., Pianesi, F., Zancanaro, M. (eds.) UMAP 2009. LNCS, vol. 5535, pp. 373–378. Springer, Heidelberg (2009)
14. Ullrich, C., Melis, E.: Pedagogically founded courseware generation based on HTN-planning. *Expert Systems with Applications* 36(5), 9319–9332 (2009)

Heuristic Multiobjective Search for Hazmat Transportation Problems^{*}

Enrique Machuca¹, Lawrence Mandow¹, José Luis Pérez de la Cruz¹,
and Antonio Iovanella²

¹ Dpto. Lenguajes y Ciencias de la Computación,
Universidad de Málaga 29071, Málaga, Spain
{machuca, lawrence, perez}@lcc.uma.es

² Dipartimento di Ingegneria dell'Impresa,
University of Rome "Tor Vergata" 00133, Rome, Italy
antonio.iovanella@uniroma2.it

Abstract. This paper describes the application of multiobjective heuristic search algorithms to the problem of hazardous material (hazmat) transportation. The selection of optimal routes inherently involves the consideration of multiple conflicting objectives. These include the minimization of risk (e.g. the exposure of the population to hazardous substances in case of accident), transportation cost, time, or distance. Multiobjective analysis is an important tool in hazmat transportation decision making. This paper evaluates the application of multiobjective heuristic search techniques to hazmat route planning. The efficiency of existing algorithms is known to depend on factors like the number of objectives and their correlations. The use of an informed multiobjective heuristic function is shown to significantly improve efficiency in problems with two and three objectives. Test problems are defined over random graphs and over a real road map.

1 Introduction

The problem of hazardous material (hazmat) transportation [7] is currently an active research topic. The search for alternative routes that minimize the risk of exposure of the population to hazardous substances can avoid bigger disasters in case of an accident. This involves the consideration of several aspects at the same time, like the transportation time, distance and cost besides risk. Multiobjective analysis [5] becomes then an important tool in hazmat transportation decision making.

In the literature, the performance of blind search multiobjective techniques has been widely analyzed [2]. In this paper, multiobjective heuristic search algorithms have been applied to the hazmat transportation problem. The experiments performed in this paper report a substantial improvement over blind multiobjective search. The cases of two and three objectives have been analyzed, achieving similar conclusions.

The paper is organized as follows. Section 2 summarizes related work on hazmat route planning and previous results in single and multiobjective search. Section 3 describes the evaluation performed while section 4 presents the experimental results.

^{*} This work is partially funded by/Este trabajo está parcialmente financiado por: Consejería de Innovación, Ciencia y Empresa. Junta de Andalucía (España), P07-TIC-03018.

A brief discussion about the results can be found in section 5. Finally, some conclusions and future work are outlined.

2 Antecedents

2.1 Hazmat Route Planning

The majority of the study on hazmat transportation deals with two related subjects: the evaluation of the risk for the population and the environment affected by the hazmat shipments, and the selection of a set of alternative paths to service the hazmat shipments. Erkut *et al.* in their survey on hazmat transportation classify routing hazmat shipments into *local* and *global* route planning problems [7]. In the local route planning problems, one is concerned with finding route(s) between a given origin-destination pair for a given hazmat, transport mode, and vehicle type. In the global route planning problem, in general, we have to find a set of paths to route hazmat shipments from distinct origins to different destinations.

There are many papers in the open literature addressing the hazmat local route planning problem. A brief summary of some key papers follows. Abkowitz and Cheng [1] developed a model that incorporates risk as a cost into a framework for optimizing the routing of hazardous materials. Kara *et al.* [13] proposed a simple modification of Dijkstra's algorithm to find a route that minimizes the exact version of the path incident probability. Erkut and Verter [8] introduced a technical risk function in which the values of risk are calculated as the product of the probability of a release accident by the consequence of the incident. Moreover, Erkut and Ingolfsson [6] proposed a simple demand satisfaction model in which the cost of multiple trips are considered in case of an incident should terminate a trip. Even if hazmat route planning is intrinsically a multi-objective problem only few papers address it by means of multi-objective optimization approaches. Cox [3] developed a multi-objective algorithm in order to find the shortest path for the hazmat transportation problem using different attributes associated to the network links, such as travel time, population density, etc; successively, Wijeratne *et al.* [20] proposed a model considering stochastic attributes for the network links. Recently, Caramia *et al.* [2] proposed an algorithm for hazmat shipments that selects k representative paths among the set of efficient paths, with respect to the minimization of length, time (cost) and risk; in particular, the selection is made by choosing paths with high spatial dissimilarity.

2.2 Multiobjective Heuristic Search

When modelling route planning as finding paths in a graph, arcs represent roads and nodes represent road junctions. Arcs labels represent road costs. We can then apply Dijkstra's [4] or A* algorithm [12]. In the case of A*, heuristic estimates are used to accelerate the search for an optimal path to a goal or destination node. While Dijkstra's algorithm uses only the accrued costs $g(n)$ of the best known paths to each node n , the selection of the next best alternative in A* is based in an evaluation function $f(n) = g(n) + h(n)$ that includes information about the estimated distance $h(n)$ from

node n to the goal. When these estimates are lower bounds of real optimal costs, A^* is guaranteed to find the optimal solutions. Additionally, under reasonable assumptions, better heuristics are known to improve search performance [17]. Several approaches have been recently proposed to improve heuristic estimates in route planning problems [9] [10].

In multiobjective graph search arcs are labelled with cost vectors. Each component in a vector represents a different attribute to be minimized. For example, in the biobjective case an arc from a node n to another m is labelled with $c(n, m) = (c_1, c_2)$. The use of cost vectors in multiobjective problems induces only a partial order relation \prec called dominance, and for all $\mathbf{v}, \mathbf{v}' \in \mathbb{R}^q$, $\mathbf{v} \prec \mathbf{v}'$ iff for all i ($1 \leq i \leq q$), $v_i \leq v'_i$ and $\mathbf{v} \neq \mathbf{v}'$ where v_i denotes the i -th component of vector \mathbf{v} . This property has an important consequence: many different nondominated paths may reach every node. Therefore, such problems are more difficult to solve than single-objective ones, since *all* nondominated solution paths must be found. This solution set is known as the *Pareto front*.

Multiobjective extensions of Dijkstra's algorithm were proposed by Hansen [11] and Martins [16]. Regarding heuristic search, three different extensions of A^* have been proposed: MOA* [18], Tung & Chew's algorithm [19], and NAMOA* [14]. The latter has been recently proven optimal among heuristic admissible multiobjective algorithms [15]. Additionally, the performance of NAMOA* has been shown to improve in a similar way to A^* with better informed heuristics. Therefore, NAMOA* is the algorithm chosen for the experiments described in this paper.

In NAMOA* each node n may be reached by a number of distinct nondominated paths (labels). Thus, all of them need to be stored, raising the memory requirements of the algorithm. These accrued costs of alternative paths are divided in two sets for each node: $G_{op}(n)$, that keeps unexplored or open labels, and $G_{cl}(n)$, that keeps explored or closed labels. Heuristic estimates can be used to speed up search. Each label $\mathbf{g} \in G_{op}(n)$ can be added to a heuristic evaluation vector $\mathbf{h}(n)$ to obtain an evaluation vector $\mathbf{f} = \mathbf{g} + \mathbf{h}(n)$, analogously to A^* .

At each iteration NAMOA* selects an open label for expansion with a nondominated \mathbf{f} -estimate. Since the lexicographic optimum of a set of vectors is known to be nondominated among them, it is frequent to select the lexicographic optimum open label for expansion. Notice that uninformed NAMOA* with lexicographic selection is equivalent to Martins' algorithm, except for some additional pruning performed by the former.

In the experiments described in this paper a precalculated multiobjective heuristic function proposed by Tung and Chew [19] is evaluated. To calculate heuristic vectors $\mathbf{h}(n) = (h_1, h_2, \dots, h_q)$ the individual h_i values are precalculated with individual single-objective Dijkstra's searches. The graph is reversed and optimal costs from the goal node to all other nodes in the graph are precomputed, once for each objective under consideration. This heuristic function can be precomputed in a practical time as single-objective searches are computationally simple compared to multiobjective search (see [19] for further details). This general and well informed precalculated heuristic has received little attention in the literature and, to the authors' knowledge, has never been evaluated in hazmat route planning.

Table 1. Correlation between pairs of objectives for both classes of problems

Problem type	Obj 1,2	Obj 1,3	Obj 2,3
Random graphs	0.01	0.01	-0.09
Lazio map	0.99	-0.18	-0.18

3 Experiments

Multiobjective heuristic search is evaluated in this paper on two different sets of problems. The first is a set of random problems with three objectives used in the work of Caramia et al [2]. The second is a set of randomly generated pairs of nodes over a real road network from the region of Lazio in Italy. This road network was also used in [2]. The experiments reported in this section evaluate the performance of blind and heuristic NAMOA* with two or three objectives.

Random Graphs. These allow the evaluation of performance depending on number of nodes, arc density and number of objectives. Different problem sets are considered with a number n of nodes equal to 100, 200 and 300. For each of these sizes, density values d were set to 0.2, 0.5 or 0.7. Each arc is labelled with a vector of three costs. Each one is an integer value in the range [1,100]. These problem sets were originally generated with a random graph generator from the 9th DIMACS Implementation Challenge on Shortest Paths¹. For the evaluation of biobjective random graphs, two of these three arc costs were selected by pairs, using the same configuration of node-density. The computation of Pearson’s correlation coefficient over pairs of two objectives is displayed in table 1. In general, these three objectives are linearly uncorrelated, resulting in rather difficult multiobjective problems. Source and goal nodes were set for all instances to 1 and n , respectively. Ten different random instances are available for each combination of n and d .

Realistic Maps. Multiobjective heuristic search is also evaluated over a real road network of the Italian region of Lazio [2]. Figure 1 shows a visual of the map with its 311 (georeferred) nodes and 879 arcs. Each arc is labelled with a vector of three costs, which represent values of distance, time and societal risk (defined as the “product between the population inside the impact zone and the incident probability”). The computation of Pearson’s correlation coefficient over pairs of two objectives can be observed in table 1. We generated a set of 50 problem instances with random source and destination nodes over this map.

Regarding the algorithms, NAMOA* was run twice for each problem instance: once without heuristic information (i.e. $\forall n \mathbf{h}(n) = \mathbf{0}$), and the second one using the pre-calculated Tung & Chew’s heuristic as described in section 2.2. Our implementation used a binary heap to implement the Open set. Only the best representative (lexicographic optimum) of each node was included in the binary heap, while the rest were kept ordered in a list at their respective nodes (as suggested in [14]). The algorithm was implemented using LispWorks Professional. The random graph problems were run

¹ <http://www.dis.uniroma1.it/~challenge9/>

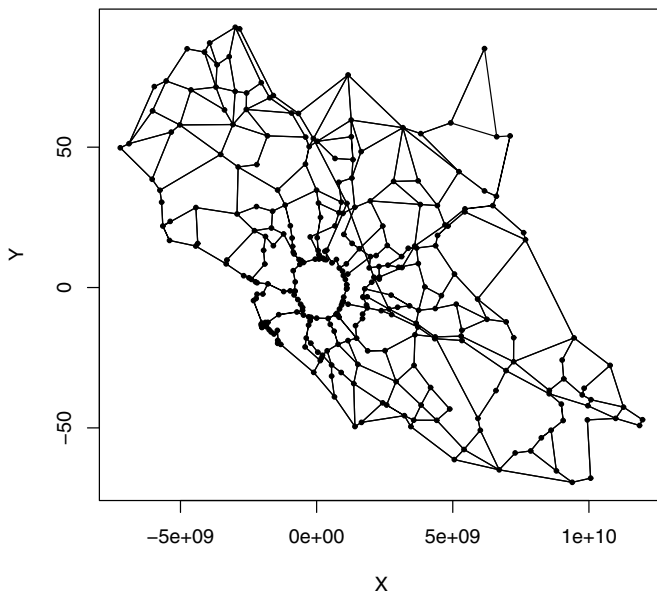


Fig. 1. Geo-referenced graph of the Lazio region in Italy

on a Windows 64-bit platform, with an Intel Core2 Quad Q9550 at 2.8Ghz, and 4Gb of RAM, and the simpler Lazio map problems on a Windows 32-bit platform, with an Intel Pentium IV and 256Mb of RAM.

4 Results

4.1 Random Graphs

Minimum, maximum and average time in seconds over the ten problems for each set of random graphs with 3 objectives were calculated. These are shown in tables 2 and 3 for blind and heuristic search respectively. The tables also report for each set the minimum, maximum and average cardinality of the set of Pareto-optimal solution costs $|C^*|$.

Analogously, two-objective search has been evaluated for all pairs of objectives. Results for blind and heuristic search with the combination 1,2 can be found in tables 4 and 5. The combination 1,3 is summarized for blind and heuristic search in tables 6 and 7 while the results for blind and heuristic search with the last combination 2,3 can be found in tables 8 and 9.

4.2 Lazio Map

Minimum, maximum and average time in seconds over the 50 problems generated for the Lazio map can be found in tables 10 and 11 for blind and heuristic search respectively. The tables report also for each of the combination of objectives (i.e. the three

Table 2. Average results on random graphs with 3 objectives for blind search

Problem class		C*			Time (Seconds)		
n	d	Min.	Avg.	Max.	Min.	Avg.	Max.
100	0.20	4	13.80	24	0.02	0.05	0.09
100	0.50	10	26.90	69	0.14	0.20	0.45
100	0.70	17	34.50	71	0.13	0.36	0.64
200	0.20	13	26.50	51	0.16	0.35	0.75
200	0.50	16	40.20	52	0.58	1.18	1.53
200	0.70	23	46.60	72	0.97	1.85	2.70
300	0.20	21	34.20	58	0.64	0.93	1.73
300	0.50	42	62.60	87	2.75	3.76	4.95
300	0.70	37	71.20	105	4.15	6.14	8.85

Table 3. Average results on random graphs with 3 objectives for heuristic search

Problem class		C*			Time (Seconds)		
n	d	Min.	Avg.	Max.	Min.	Avg.	Max.
100	0.20	4	13.80	24	0.00	0.01	0.02
100	0.50	10	26.90	69	0.01	0.05	0.17
100	0.70	17	34.50	71	0.01	0.11	0.36
200	0.20	13	26.50	51	0.02	0.07	0.17
200	0.50	16	40.20	52	0.17	0.32	0.42
200	0.70	23	46.60	72	0.11	0.57	0.94
300	0.20	21	34.20	58	0.08	0.18	0.42
300	0.50	42	62.60	87	0.83	1.07	1.53
300	0.70	37	71.20	105	1.19	1.92	3.14

Table 4. Average results on random graphs with objectives 1,2 for blind search

Problem class		C*			Time (Seconds)		
n	d	Min.	Avg.	Max.	Min.	Avg.	Max.
100	0.20	2	4.80	7	0.00	0.02	0.03
100	0.50	5	8.60	15	0.03	0.06	0.13
100	0.70	4	9.20	13	0.03	0.09	0.14
200	0.20	3	8.60	16	0.03	0.11	0.17
200	0.50	5	8.70	13	0.16	0.24	0.33
200	0.70	7	10.30	15	0.27	0.35	0.42
300	0.20	6	10.70	15	0.09	0.25	0.41
300	0.50	6	12.30	20	0.44	0.66	1.03
300	0.70	7	12.10	16	0.56	0.90	1.17

objectives at the same time, the 1st-2nd, the 1st-3rd and 2nd-3rd) the minimum, maximum and average cardinality of the $|C^*|$ set of Pareto-optimal solution costs reported by the algorithm, as done with the random graphs.

Table 5. Average results on random graphs with objectives 1,2 for heuristic search

Problem class		C^*			Time (Seconds)		
n	d	Min.	Avg.	Max.	Min.	Avg.	Max.
100	0.20	2	4.80	7	0.00	0.00	0.02
100	0.50	5	8.60	15	0.00	0.02	0.03
100	0.70	4	9.20	13	0.02	0.03	0.06
200	0.20	3	8.60	16	0.00	0.02	0.05
200	0.50	5	8.70	13	0.01	0.05	0.08
200	0.70	7	10.30	15	0.05	0.08	0.14
300	0.20	6	10.70	15	0.01	0.04	0.09
300	0.50	6	12.30	20	0.06	0.15	0.30
300	0.70	7	12.10	16	0.08	0.19	0.41

Table 6. Average results on random graphs with objectives 1,3 for blind search

Problem class		C^*			Time (Seconds)		
n	d	Min.	Avg.	Max.	Min.	Avg.	Max.
100	0.20	1	4.80	7	0.00	0.02	0.05
100	0.50	4	8.00	16	0.02	0.05	0.09
100	0.70	5	9.10	13	0.03	0.08	0.13
200	0.20	5	7.80	13	0.03	0.09	0.16
200	0.50	5	9.10	19	0.13	0.22	0.39
200	0.70	7	9.10	13	0.22	0.31	0.45
300	0.20	6	9.90	17	0.09	0.22	0.33
300	0.50	9	11.80	16	0.33	0.62	1.01
300	0.70	7	12.30	16	0.59	0.92	1.25

Table 7. Average results on random graphs with objectives 1,3 for heuristic search

Problem class		C^*			Time (Seconds)		
n	d	Min.	Avg.	Max.	Min.	Avg.	Max.
100	0.20	1	4.80	7	0.00	0.01	0.02
100	0.50	4	8.00	16	0.00	0.01	0.03
100	0.70	5	9.10	13	0.00	0.02	0.05
200	0.20	5	7.80	13	0.00	0.01	0.03
200	0.50	5	9.10	19	0.01	0.04	0.13
200	0.70	7	9.10	13	0.03	0.06	0.09
300	0.20	6	9.90	17	0.01	0.05	0.11
300	0.50	9	11.80	16	0.05	0.14	0.25
300	0.70	7	12.30	16	0.11	0.21	0.44

5 Discussion

The comparison of tables 2-3, 4-5, 6-7, 8-9 and 10-11 shows that heuristic estimates led the search more quickly to optimal solutions in all cases presented in the paper. In the

Table 8. Average results on random graphs with objectives 2,3 for blind search

Problem class		C^*			Time (Seconds)		
n	d	Min.	Avg.	Max.	Min.	Avg.	Max.
100	0.20	3	5.70	9	0.00	0.02	0.03
100	0.50	3	8.30	12	0.03	0.06	0.08
100	0.70	7	10.40	15	0.06	0.09	0.13
200	0.20	3	7.80	11	0.05	0.09	0.16
200	0.50	6	10.00	17	0.11	0.25	0.47
200	0.70	4	9.80	21	0.11	0.35	0.62
300	0.20	5	10.00	14	0.14	0.23	0.33
300	0.50	5	11.70	17	0.30	0.64	0.98
300	0.70	8	11.70	20	0.61	0.89	1.59

Table 9. Average results on random graphs with objectives 2,3 for heuristic search

Problem class		C^*			Time (Seconds)		
n	d	Min.	Avg.	Max.	Min.	Avg.	Max.
100	0.20	3	5.70	9	0.00	0.01	0.02
100	0.50	3	8.30	12	0.00	0.02	0.03
100	0.70	7	10.40	15	0.00	0.03	0.08
200	0.20	3	7.80	11	0.00	0.02	0.03
200	0.50	6	10.00	17	0.02	0.07	0.14
200	0.70	4	9.80	21	0.01	0.09	0.20
300	0.20	5	10.00	14	0.00	0.04	0.06
300	0.50	5	11.70	17	0.05	0.13	0.30
300	0.70	8	11.70	20	0.08	0.18	0.33

Table 10. Average results on Lazio map for blind search

Problem class		C^*			Time (Seconds)		
$ obj $	obj	Min.	Avg.	Max.	Min.	Avg.	Max.
3	1,2,3	1	3.96	20	0.00	0.17	0.77
2	1,2	1	1.06	2	0.00	0.03	0.07
2	1,3	1	3.92	18	0.00	0.17	0.73
2	2,3	1	3.36	14	0.00	0.14	0.63

Table 11. Average results on Lazio map for heuristic search

Problem class		C^*			Time (Seconds)		
$ obj $	obj	Min.	Avg.	Max.	Min.	Avg.	Max.
3	1,2,3	1	3.96	20	0.00	0.02	0.17
2	1,2	1	1.06	2	0.00	0.01	0.02
2	1,3	1	3.92	18	0.00	0.03	0.15
2	2,3	1	3.36	14	0.00	0.02	0.11

class of random graphs, NAMOA* combined with Tung and Chew's heuristic is always several times faster on average than blind search (from 3.95 times in biobjective (1,3) to 4.87 times in (2,3) combination).

The tables also show that time devoted to find a shortest path in a graph increases with the density and the number of nodes of the graph.

Considering random graphs, all the biobjective pairs (1,2), (1,3) and (2,3) (tables 4-5, 6-7 and 8-9) present very similar results. On the other hand, time is greater for three objective problems (tables 2-3).

Regarding the Lazio map problems, the analysis shows a different behaviour. As seen in tables 10 and 11, biobjective problems for the pair (1,2) are easier, while biobjective problems for the pairs (1,3) and (2,3) are more difficult. Perhaps surprisingly, three-objective problems have no additional difficulty over biobjective problems (1,3) and (2,3).

The explanation of this phenomenon can be found in the analysis of correlation between costs. In the case of random costs, the correlation is very low for every pair of objectives; therefore, for the same density and size, difficulty is similar for any two objectives, and greater for the set of three objectives.

However, correlation between objectives in Lazio maps depends on the pair considered (table 1). Time and distance are highly correlated ($\rho \approx 1$), but societal risk is not linearly correlated with any of them ($\rho \approx 0$). Therefore the number of Pareto-optimal solutions is not affected by considering objective 2 if objective 1 has been considered.

In the work of Caramia et al. [2] similar information is shown only for blind search (Martins' algorithm) over the three-objective case in the random graph set. Solution times are faster in the present paper even for blind NAMOA* search. These differences can be attributed to different pruning and implementation schemes.

6 Conclusions and Future Work

The paper presents an analysis of blind and heuristic search for multiobjective hazmat transportation problems. The analysis involves the consideration of two and three objectives over two classes of problems: random graphs and hazmat transportation problems defined over a real map from the Lazio region in Italy (in this case, the objectives involved are distance, travel time and societal risk).

From the systematic evaluation of several parameters performed in the paper some conclusions about multiobjective search can be drawn. Concerning the use of heuristics, heuristic estimates allowed faster searches for all cases presented in the paper. As expected, problem difficulty increases with graph size and node density. The paper also shows the importance of correlation between objectives in the cases considered. The number of Pareto-optimal paths falls as the correlation between objectives increases. Thus, the time needed to solve a multiobjective problem depends on the specific nature of arc costs. In problems with relatively uncorrelated objectives the number of Pareto-optimal paths increases with the number of objectives under consideration, while the addition of highly correlated objectives does not degrade the performance of the search.

Future work will consider the application of multiobjective heuristic search to larger hazmat road maps, and the development of more efficient heuristic functions.

A formal analysis on the influence of the number of objectives and their correlation in the performance of search algorithms is also of great interest.

References

1. Abkowitz, M., Cheng, P.D.: Developing a risk/cost framework for routing truck movements of hazardous materials. *Accident Analysis & Prevention* 20(1), 39–51 (1988)
2. Caramia, M., Giordani, S., Iovanella, A.: On the selection of k routes in multiobjective hazmat route planning. *IMA Journal of Management Mathematics* 21, 239–251 (2010)
3. Cox, R.G.: Routing and scheduling of hazardous materials shipments: algorithmic approaches to managing spent nuclear fuel transport. Ph.D. thesis, Cornell University, Ithaca, NY (1984)
4. Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 269–271 (1959)
5. Ehrgott, M.: *Multicriteria Optimization*. Springer, Heidelberg (2005)
6. Erkut, E., Ingolfsson, A.: Transport risk models for hazardous materials: revisited. *Operations Research Letters* 33(1), 81–89 (2005)
7. Erkut, E., Tjandra, S.A., Verter, V.: Hazardous Materials Transportation. In: Barnhart, C., Laporte, G. (eds.) *Handbook in OR and MS*, vol. 14, pp. 539–621. Elsevier (2007)
8. Erkut, E., Verter, V.: Modeling of transport risk for hazardous materials. *Operations Research* 46(5), 625–642 (1998)
9. Goldberg, A.V., Harrelson, C.: Computing the shortest path: A^* search meets graph theory. In: *SODA 2005 Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 156–165 (2005)
10. Goldberg, A.V., Kaplan, H., Werneck, R.F.F.: Better Landmarks Within Reach. In: Demetrescu, C. (ed.) *WEA 2007. LNCS*, vol. 4525, pp. 38–51. Springer, Heidelberg (2007)
11. Hansen, P.: Bicriterion path problems. *Lecture Notes in Economics and Mathematical Systems*, vol. 177, pp. 109–127. Springer, Heidelberg (1979)
12. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Systems Science and Cybernetics* SSC-4(2), 100–107 (1968)
13. Kara, B.Y., Erkut, E., Verter, V.: Accurate calculation of hazardous materials transport risks. *Operations Research Letters* 31(4), 285–292 (2003)
14. Mandow, L., Pérez de la Cruz, J.L.: A new approach to multiobjective A^* search. In: *Proc. of the XIX Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pp. 218–223 (2005)
15. Mandow, L., Pérez de la Cruz, J.L.: Multiobjective A^* search with consistent heuristics. *Journal of the ACM* 57(5), 27:1–27:25 (2010)
16. Martins, E.: On a multicriteria shortest path problem. *European Journal of Operational Research* 16, 236–245 (1984)
17. Pearl, J.: *Heuristics*. Addison-Wesley, Reading (1984)
18. Stewart, B.S., White, C.C.: Multiobjective A^* . *Journal of the ACM* 38(4), 775–814 (1991)
19. Tung, C.T., Chew, K.L.: A multicriteria Pareto-optimal path algorithm. *European Journal of Operational Research* 62, 203–209 (1992)
20. Wijeratne, A.B., Turnquist, M.A., Mirchandani, P.B.: Multiobjective routing of hazardous materials in stochastic networks. *European Journal of Operational Research* 65(1), 33–43 (1993)

Topography of Functional Connectivity in Human Multichannel Electroencephalogram during Second Language Processing

Ernesto Pereda^{1,*}, Susanne Reiterer^{2,3}, and Joydeep Bhattacharya^{4,5}

¹ Department of Basic Physics, University of La Laguna, Tenerife, Spain
eperdepa@ull.es

² Department of Neuroradiology, Section of Experimental MR of the CNS,
University of Tübingen, Germany

³ Hertie Institute for Clinical Brain Research, General Neurology,
University of Tübingen, Germany

susanne.reiterer@med.uni-tuebingen.de

⁴ Department of Psychology, Goldsmiths College, University of London,
London, United Kingdom

⁵ Commission for Scientific Visualization, Austrian Academy of Sciences, Vienna, Austria
j.bhattacharya@gold.ac.uk

Abstract. We analyze the topography of nonlinear functional connectivity in the EEG of two groups of German-native speakers, divided according to their English proficiency level (high or low), when listening to one text in German and one in English. Global interdependence was assessed in full-band EEG by means of an index of multivariate correlation derived from the normalized cross-mutual information between every two electrodes within each region of interest (ROI): three interhemispheric (frontal, centro-temporal and parieto-occipital) and two intrahemispheric ones (left and right hemisphere). The results show clear topographic differences between the interhemispheric ROIs, but no differences between the intrahemispheric ROIs. Furthermore, there are also differences in language processing that depend on the proficiency level. We discuss these results and their implication along with recent findings on phase synchronization in the gamma band during second language processing.

Keywords: EEG, second language processing, functional connectivity, joint entropy.

1 Introduction

Non-linear multivariate time series analysis has been extensively and successfully used during the last decade to study brain dynamics from EEG and MEG records in different situations (see, e.g., [1] for a review). Indeed, the term *functional*

* Corresponding author.

*connectivity*¹ has been coined to refer to the existence of statistical dependencies between signals recorded from distinct units (ranging from single neurons to whole brain areas) within a nervous system [2]. Initial works in this line of research were concerned with the analysis of the statistical interdependence between two units using bivariate nonlinear indexes of, e.g., generalized or phase synchronization. However, with more and more experiments simultaneously recording an increasing number of sites, it became apparent that we need new, truly multivariate approaches that allow the characterization of the collective dynamics of more than two units [3-5]. We have recently used one of these approaches to characterize the global phase synchronization in the gamma band of the EEG during second language processing and its dependence on the proficiency level of the subjects [6]. In this work, we complement and extend our earlier result by studying the topography of the functional connectivity during second language processing of Full-band EEG by making use of an index of nonlinear correlation based on the concept of Mutual information [5].

2 Material and Methods

2.1 Groups of Subjects

The two groups of subjects contrasted have been described elsewhere [6], thus we only describe them briefly here. Thirty eight university students with comparable educational levels were divided into two groups of 19 subjects each according to their second language proficiency level (L2 = English). The ‘high proficiency group’ subjects (HP) were advanced university language students studying English language and linguistics for a master’s degree (last year, 5-6 years completed). Their level of English proficiency was ‘‘very good’’ (so-called ‘‘native speaker-like’’ performance) or ‘‘good’’ according to their performances at university: they all had high levels of linguistic training and knowledge at the time of experiment. Additionally, they had spent abroad in an English speaking country an average of 10 months. By contrast, subjects in the ‘low-proficiency group’ (LP) were university students of various disciplines other than English language and linguistics. They displayed medium to low level of second language skills (corresponding to the three rating-system groups ‘‘medium’’, ‘‘lower-level’’ and ‘‘lowest-level’’), which were sufficient to let them pass their school leaving exams (‘‘Matura’’, an equivalent to ‘‘A levels’’), but since then were not developed any further. They were able to lead basic level conversations in English, but their speech was non-fluent. The average amount of time LP subjects spent abroad in an English speaking country was 5 weeks. With regard to the country where they had spent some time, the groups were homogeneous.

All subjects were right-handed (measured by the Edinburgh handedness inventory) female students with German as their native language. We rigidly controlled for the variable gender in order to avoid possible influences of gender onto the processing of

¹ The definition of functional connectivity given here, is the most commonly accepted nowadays, although a search in Google of the expression ‘‘functional connectivity’’ produces, as of May 1st, 2011, no less than 190.000 results, some of them with different definitions of this concept.

language and its neural representations. Mean (SD) age was 24 years (2.3 years and 2.7 years respectively for two groups) for both groups. They were also matched for socio-cultural background and education: all subjects had similar social (middle class), educational (university students), and cultural (living in Vienna) background.

We strictly controlled for the variable “age of onset” of L2 learning. The average (\pm s.d.) age of onset was 9 yrs (1 yr) and was matched between the two groups. The controlled variables were: age, handedness, gender, mother tongue, socio-educational and cultural background and region of residence.

The study was in compliant with the Code of Ethics of the World Medical association (Declaration of Helsinki) and the experimental protocol was approved by the local ethics committee. All subjects gave their written informed consent for the study.

2.2 Stimulus

We used coherent spoken speech (radio news) as stimuli in a listening comprehension and discourse processing paradigm. In cooperation with the English department at the University of Vienna, the speech samples were matched for syntactic complexity, semantic contents & genre, discourse structure and gender of the speaker (all male speakers). Within the framework of a block design, six blocks of coherent speech (2.0 – 3.2 min each) with randomly inserted baseline blocks (acoustic noise, 2.0 min each) were presented in randomized order: three blocks in condition L2 English and three blocks in condition L1 German were auditorily presented in randomized order over earphones.

2.3 Data Recording and Pre-processing

We recorded multivariate EEG signals during L1 and L2 processing in a quiet, dimly-lit sound-proof experimental room. subjects were monitored through a video control system during the recording session in order to control for possible movements. Nineteen gold-disc electrodes were carefully attached to the scalp with adhesive electrode gel, positioned according to the international 10/20 System (Fig. 1); one additional frontal electrode was used as a ground, and two separate electrodes, at the right and left ear-lobe, were used as reference electrodes. The recordings were re-referenced against the algebraic mean of the two ear-lobe electrodes [7]. Eye movements were additionally controlled for by a piezo-electric device attached to the eyelid. Using a conventional Nihon-Kohden 21 channel recorder, the EEG was amplified, filtered (time constant 0.3 s.), displayed and recorded at a sampling rate of 128 Hz. Electrode’s impedance was kept below 5 k Ω . A notch filter at 50 Hz was used for the elimination of power line contamination. Finally, we rejected those epochs containing samples of voltages higher than 70 μ V (absolute value), plus additional epochs where 2% or more samples deviated more than 3 standard deviations from the mean value.

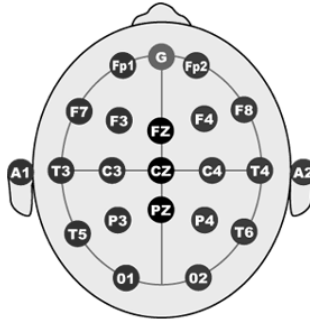


Fig. 1. Placement of the 19 electrodes recorded in the study (G represents the ground electrode; A1 and A2 are the linked earlobes used as reference)

2.4 Data Analysis

2.4.1 Assessment of Multivariate Functional Connectivity: The Nonlinear Correlation Information Entropy, I_R

In order to assess functional connectivity in our multivariate data set, we made use of the so-called nonlinear correlation information entropy [5], whose calculation is outlined henceforth.

Given two discrete variables $X=[x_i]_{i=1,\dots,N_s}$ and $Y=[y_i]_{i=1,\dots,N_s}$, from which N_s samples have been obtained, we first sort, in ascending order, these samples and bin them into b ranks, with the first N_s/b samples of each variable placed in the first rank, the second N_s/b samples placed in the second rank, and so on. Then, the sample pairs $[(x_i,y_i)]_{i=1,\dots,N_s}$ are placed into a $b \times b$ rank grids by comparing each sample pair to the rank sequences of X and Y . The revised entropy of X is defined as:

$$H^r(X) = - \sum_{i=1}^b \frac{n_i}{N_s} \log_b \frac{n_i}{N_s} \tag{1}$$

and the revised joint entropy of the two variables X and Y :

$$H^r(X,Y) = - \sum_{i=1}^b \sum_{j=1}^b \frac{n_{ij}}{N_s} \log_b \frac{n_{ij}}{N_s} \tag{2}$$

where n_{ij} is the number of samples in the ij th rank grid. The nonlinear correlation coefficient $NCC(X;Y)$ is:

$$NCC(X;Y) = H^r(X) + H^r(Y) - H^r(X,Y) \tag{3}$$

where $H^r(Y)$ is defined in complete analogy with (1). Due to the binning scheme, n_i is invariant for both X and Y and equal to N_s/b . Thus, $NCC(X;Y)$ reduces to:

$$NCC(X; Y) = 2 + \sum_{i=1}^{b^2} \frac{n_{ij}}{N_s} \log_b \frac{n_{ij}}{N_s} \tag{4}$$

If the sample sequences are exactly the same, the last right-hand side term of the above equation equals -1 and thus, $NCC(X;Y)=1$, whereas if the two variables are completely uncorrelated, the sample pairs distribute equally into the $b \times b$ ranks, the sum equals to -2 and $NCC(X;Y)=0$.

In the case of $k>2$ variables (e.g., more than two EEG channels), we obtain a symmetric squared $k \times k$ matrix of nonlinear correlation coefficients:

$$R = \{NCC_{ij}\}_{i,j=1,\dots,k} \tag{5}$$

where NCC_{ij} is the nonlinear correlation coefficient between signals i and j , and $NCC_{ij}=NCC_{ji}$. Besides, $NCC_{ij} = 1$ if $i=j$, and $0 \leq NCC_{ij} \leq 1$ when $i \neq j$. Thus, R is a Hermitian matrix, which is also positive semidefinite. The sum of its eigenvalues equals the trace, i.e.:

$$\sum_{n=1}^k \lambda_n = k \tag{6}$$

Recent studies on multivariate EEG analysis have taken advantage of the spectral properties of this kind of matrixes (see, e.g., [4, 8] for the equal time correlation matrix, and [3] for the bivariate phase synchronization matrix). Additionally, the study of the corresponding eigenvectors matrixes makes it possible to define a participation index that assigns each electrode to a given cluster [3]. The underlying idea is easy to understand if we analyze the two extreme cases of k completely correlated and k completely independent signals. In the first case, $NCC_{ij} = 1, \forall i, j = 1, \dots, k$, and $\lambda_1=k, \lambda_n=0 (n=2,\dots,k)$. Conversely, in the second one, $NCC_{ij} = 0$ whenever $i \neq j$, and $\lambda_n=1 (n=1,\dots,k)$. In a (more realistic) intermediate case, a subgroup of the higher eigenvalues, which characterize dynamical clusters of functionally connected EEG channels, is greater than 1, whereas the rest are lower than 1.

In the case of (5), this spectral property can be used to define a nonlinear index of multivariate correlation among $k>2$ signals. The *nonlinear joint entropy* of the k variables is derived from R as follows:

$$H_R = - \sum_{i=1}^k \frac{\lambda_i}{k} \log_k \frac{\lambda_i}{k} \tag{7}$$

From the behaviour of the eigenvalues explained above, it follows immediately that (7) is 0 if the k signals are completely correlated and 1 if they are completely independent.

Thus, I_R , defined as:

$$I_R = \mathbf{1} - H_R = \mathbf{1} + \sum_{i=1}^k \frac{\lambda_i}{k} \log_k \frac{\lambda_i}{k} \quad (8)$$

is an index of multivariate nonlinear correlation among k signals (termed the *nonlinear correlation information entropy*), which equals 1 if they are completely correlated, and 0 if they are completely independent. In an intermediate case, one has $0 < I_R < 1$, with the index closer to 1 the more correlated are the signals. Note that, since (8) is obtained from the eigenvalues of R (whose elements are nonlinear correlation indexes), I_R is sensitive to both linear and nonlinear correlations among the k signals. This represents an advantage of I_R over similar indexes such as the one described in [4], which are only sensitive to linear correlations.

2.4.2 Regions of Interests

We studied the topography of functional connectivity in both groups of subjects and both conditions by defining three different, non-overlapping interhemispheric regions of interest (ROIs): frontal region (FR), which includes electrodes Fp1, Fp2, F7, F3, Fz, F4 and F8; centro-temporal region (CT), which includes electrodes T3, C3, Cz, C4, T4, T5 and T6; parieto-occipital region (PO), which includes electrodes P3, Pz, P4, O1 and O2; and two intrahemispheric ROIs: left hemisphere (LH), including electrodes Fp1, F7, F3, C3, T3, C3, P3 and O1, right hemisphere (RH), including electrodes Fp2, F4, F8, C4, T4, P4, T6, and O2.

2.4.3 Practical Aspects of I_R Calculation

One practical issue that is necessary to deal with when estimating I_R is that the fact that it is a *parametric* index, i.e., it depends on two parameters: the number of data samples, N_s , and the number of ranks, b . Typically, entropy estimations based on data binning may be strongly biased if either the total number of data or the average number of data in each bin are not long enough, which gives rise to high entropy values (see, e.g., [9], for a review of entropy estimation methods from data samples). Thus, it is necessary to determine a priori which are suitable values of both parameters to avoid (or at least, reduce as much as possible) such overestimation. Fig. 2 exemplifies, using the FR region of one subject, the typical behavior of I_R as a function of b and N_s . As can be seen for the figure, lower values of N_s and high values of b tend to produce high values of I_R . According to this result, we took $N_s=4000$ (which correspond to 39.6 s) and $b=20$.

Thus, for every subject and ROI, we slid a moving window of size N_s along the whole record, and calculate I_R as the average of this index for the N_W windows²:

$$I_R = \frac{1}{N_W} \sum_{i=1}^{N_W} I_{R_i} \quad (9)$$

² The Matlab[®] script to calculate I_R is available upon request from the corresponding author.

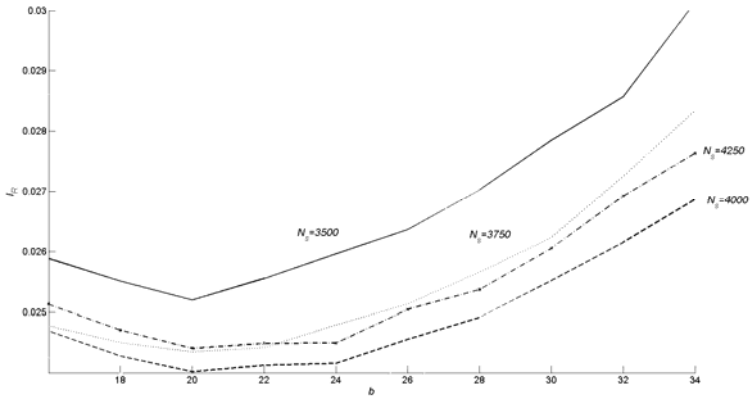


Fig. 2. I_R as a function of N_s and b for the FR region of one subject

2.5 Statistical Comparisons

A multivariate analysis of the variance (MANOVA) test was used to check for the existence of between- and within- group differences in the average I_R . Thus, interhemispheric ROIs were studied with proficiency (H vs. L) as between group factor and Language (L1 vs. L2) and region (FR, CT and PO) as dependent factors. Likewise, intrahemispheric ROIs were studied by substituting in the above scheme the three intrahemispheric ROIs by the two interhemispheric ones. We used the conservative Bonferroni post-hoc test, when appropriate, to get further insight into the origin of these differences, which were considered significant for $p < 0.05$.

As an additional precaution against false positives, we used a Levene's test to check the homogeneity of the variances of the different groups before applying the MANOVA test. All the statistical calculations were carried out using the data analysis software system STATISTICA³ (StatSoft, Inc. (2008)) version 8.0.

3 Results

The Levene's test was not significant for either the interhemispheric or the intrahemispheric ROIs analysis, which indicates that the variance is homogeneous in all cases.

Figure 3 presents the results corresponding to the interhemispheric ROIs, which can be summarized as follows: there are global within-group differences among ROIs ($p < 0.001$), with a lower I_R for the CT region than for the other two regardless of the language and the proficiency, and a further increase of the index in the PO region for L2 (both groups) as well as L1 (HP group). Furthermore, I_R was lower for L2 as compared to L1 for the LP group in the FR region.

³ <http://www.statsoft.com>

The results for the two intrahemispheric ROIs are shown in figure 4. In this case, there are neither between- nor within-group differences.

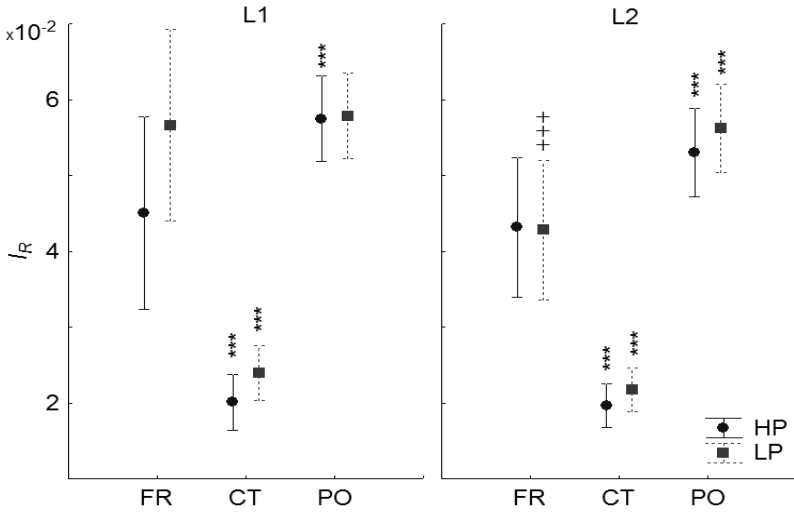


Fig. 3. Average I_R for the three interhemispheric ROIs (FR: frontal, CT: centrotemporal; PO: parieto-occipital) and both proficiency groups (HP: high proficiency, LP: Low proficiency) during L1 (left) and L2 processing (right). Vertical bars denote 0,95 confidence intervals. Asterisks stand for within-group regional differences (vs. CT). Crosses stand for L1 vs L2 differences. ***,+++: $p < 0.001$ (Bonferroni post-hoc test).

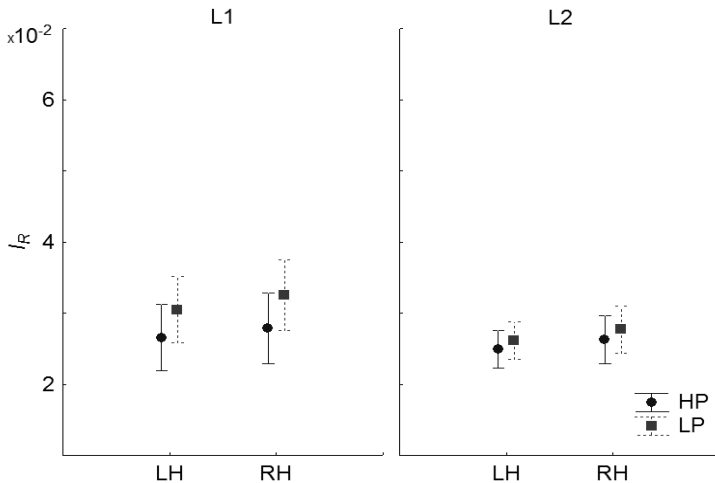


Fig. 4. Same as in Fig. 3 but for the intrahemispheric ROIs (LH: left hemisphere; RH: right hemisphere). We use the same upper and lower limits for the vertical axis as in Fig. 3 for comparability.

4 Discussion

We have shown in this work that functional EEG connectivity during language processing presents topographic interhemispheric (but not intrahemispheric) differences, with the FR and the PO regions showing greater collective cooperation than the CT region. Additionally, although we did not directly compare them, as it is apparent from fig. 3 and 4 the two former region of interests presented greater functional connectivity than the two intrahemispheric region of interests, indicating that cooperation within FR and PO regions is superior to that within the two hemispheres.

As for differences between L1 and L2 processing, they were found only in the FR region. Moreover, whereas the topographic differences are the same for the two groups, differences in language processing exist only for the LP group, where functional connectivity in the FR region decreases during L2 processing.

A straightforward conclusion of this latter result would be that L2 proficiency correlates with the degree of frontal functional connectivity, because native-like L2 proficiency gives rise to a functional interhemispheric integration in this region that is essentially equal to that found during L1 processing. Yet in an earlier work we found that gamma band phase synchronization is greater in LP than in HP subjects during L2 processing [6], which we explained within the framework of the cortical efficiency hypothesis. According to it, persons who are good at a certain task use a limited group of brain circuits or use their neuronal subroutines more efficiently, thus requiring fewer neuronal networks to accomplish a task, while poor performers (for whom problems are hard) use more circuits, which are inessential or inefficient for task performance and this is reflected in higher overall patterns of activity [10].

Taken together, past and present results on the relationship between functional EEG connectivity and L2 proficiency level suggest that high proficiency, native-like, processing of L2 is carried out with the same balance between functional segregation and integration that is present during L1 processing. However, L2 low-proficiency level produces an increase of full-band frontal functional segregation and gamma band functional integration. This increase in gamma band functional connectivity may be therefore a mechanism to compensate the reduced cooperation during L2 processing (as assessed by I_R) among the frontal areas of LP subjects in the full-band EEG. In contrast, HP subjects, who process L2 almost automatically, do it thanks to the proficient cooperation of their frontal areas. Thus, careful analysis of functional connectivity of full-band EEG is necessary to thoroughly characterize, on the one hand, the balance between integration and segregation of the brain areas that participate in L2 processing; and, on the other hand, the changes in functional connectivity that distinguishes LP from HP subjects.

References

1. Pereda, E., Quian Quiroga, R., Bhattacharya, J.: Nonlinear multivariate analysis of neurophysiological signals. *Prog. Neurobiol.* 77, 1–37 (2005)
2. Friston, K.J.: Functional and effective connectivity in neuroimaging: a synthesis. *Hum. Brain Mapp.* 2, 56–78 (1994)

3. Allefeld, C., Muler, M., Kurths, J.: Eigenvalue decomposition as a generalized synchronization cluster analysis. *Int. J. Bifurcat. Chaos* 17, 3493–3497 (2007)
4. Müller, M., Baier, G., Galka, A., Stephani, U., Muhle, H.: Detection and characterization of changes of the correlation structure in multivariate time series. *Phys. Rev. E* 71, 046116 (2005)
5. Wang, Q., Shen, Y., Zhang, J.Q.: A nonlinear correlation measure for multivariable data set. *Physica D* 200, 287–295 (2005)
6. Reiterer, S., Pereda, E., Bhattacharya, J.: Measuring second language proficiency with EEG synchronization: how functional cortical networks and hemispheric involvement differ as a function of proficiency level in second language speakers. *Second Language Research* 25, 77–106 (2009)
7. Essl, M., Rappelsberger, P.: EEG coherence and reference signals: experimental results and mathematical explanations. *Med. Biol. Eng. Comput.* 36, 399–406 (1998)
8. Rummel, C., Baier, G., Muller, M.: Automated detection of time-dependent cross-correlation clusters in nonstationary time series. *Europhys. Lett.* 80, 68004 (2007)
9. Hlavackova-Schindler, K., Palus, M., Vejmelka, M., Bhattacharya, J.: Causality detection based on information-theoretic approaches in time series analysis. *Phys. Rep.* 441, 1–46 (2007)
10. Haier, R.J., Siegel Jr., B.V., MacLachlan, A., Soderling, E., Lottenberg, S., Buchsbaum, M.S.: Regional glucose metabolic changes after learning a complex visuospatial/motor task: a positron emission tomographic study. *Brain Res.* 570, 134–143 (1992)

A Summary on the Study of the Medium-Term Forecasting of the Extra-Virgen Olive Oil Price

Antonio Jesús Rivera, María Dolores Pérez-Godoy, María José del Jesus,
Pedro Pérez-Recuerda, María Pilar Frías, and Manuel Parras

University of Jaén, Spain

{arivera,lperez,mjjesus,pedro.perez.recuerda,mpfrias,mparras}@ujaen.es

Abstract. In this paper we present a summary of the application of CO²RBFN, a evolutionary cooperative-competitive algorithm for Radial Basis Function Networks design, to the medium-term forecasting of the extra-virgen olive price, carry out by the SIMIDAT research group. The forecast is about the price at source of the extra-virgin olive oil six months ahead. The influential of the feature selection algorithms over the forecasting of the extra-virgin olive oil price has been analysed in this study and the results obtained with CO²RBFN have been compared with those obtained by different soft computing methods.

Keywords: times series forecasting, olive oil, RBFN, technical indicator.

1 Introduction

Nowadays, olive oil is an important business sector in an expanding market and Spain is the first producer and exporter. The Official Market for the negotiation of future contracts for olive oil (MFAO)¹ in Spain is a society whose objective is to forecast prices to balance supply and demand in future periods of time. The aim of this work is to predict these future prices in order to increase the global benefits of the sector. Agents of the olive oil sector consider more important a medium-term prediction than a short-term prediction of the olive oil price, specially for the Official Market for the negotiation of future contracts for olive oil.

Authors have developed an algorithm for the cooperative-competitive design of Radial Basis Functions Networks, CO²RBFN, that has been successfully used in short-term forecasting of time series [11] [12]. In this paper we present a summary of the study [12] carry out by the author with the objective of forecasting the price at source of the extra-virgin olive oil six months ahead. To help in this task the price itself as well as up to 9 exogenous variables (such as price at destination, opening and closing stock, consumer price index, etc) have been taken into account. With the aim of preprocessing these input variables, technical indicators such as momentums, oscillators, disparities, etc. are used. Due to the combination of technical indicators and exogenous variables a high number of input variables are obtained. Therefore, the application of feature selection

¹ <http://www.mfao.es>

algorithms is also analyzed in order to determine the more influential variables in the forecasting of the extra-virgin olive oil price.

The rest of the paper is organized as follows: section 2 CO²RBFN applied to time series forecasting is detailed. The experimental framework is described in section 3. The results obtained for the forecasting methods used are detailed in Section 4. In Section 5, conclusions and future work are outlined.

2 CO²RBFN for Time Series Forecasting

CO²RBFN [12], is an hybrid evolutionary cooperative-competitive algorithm for the design of RBFNs. In this algorithm each individual of the population corresponds, using a real representation, to an RBF and the entire population is responsible for the final solution. The individuals cooperate towards a definitive solution, but they must also compete for survival.

In this environment, in which the solution depends on the behavior of many components, the fitness of each individual is known as credit assignment. In order to measure the credit assignment of an individual, three factors have been proposed: the RBF contribution to the network output, the error in the basis function radius and the degree of overlapping among RBFs.

The application of the operators is determined by a Fuzzy Rule-Based System. The inputs of this system are the three parameters used for credit assignment and the outputs are the operators' application probability.

The main steps of CO²RBFN, explained in the following subsections, are shown in Figure 1 in pseudocode.

1. Initialize RBFN
2. Train RBFN
3. Evaluate RBFs
4. Apply operators to RBFs
5. Substitute the eliminated RBFs
6. Select the best RBFs
7. If the stop condition is not verified go to step 2

Fig. 1. Main steps of CO²RBFN

RBFN Initialization. To define the initial network, a specified number m of neurons (i.e. the size of population) is randomly allocated among the different patterns of the training set.

RBFN Training. The Least Mean Square algorithm [14] has been used to calculate the RBF weights.

RBF Evaluation. A credit assignment mechanism is required in order to evaluate the role of each RBF ϕ_i in the cooperative-competitive environment. For an RBF, three parameters, a_i , e_i , o_i are defined:

- The contribution, a_i , of the RBF ϕ_i , $i = 1 \dots m$, is determined by considering the weight, w_i , and the number of patterns of the training set inside its width, p_i :

$$a_i = \begin{cases} |w_i| & \text{if } p_i > q \\ |w_i| * (p_i/q) & \text{otherwise} \end{cases} \tag{1}$$

where q is the average of the p_i values minus the standard deviation of the p_i values.

- The error measure, e_i , for each RBF ϕ_i , is obtained by calculating the Mean Absolute Percentage Error (MAPE) inside its width:

$$e_i = \frac{\sum_{\forall p_i} \left| \frac{f(p_i) - y(p_i)}{f(p_i)} \right|}{np_i} \tag{2}$$

where $f(p_i)$ is the output of the model for the point p_i , inside the width of RBF ϕ_i , $y(p_i)$ is the real output at the same point, and np_i is the number of points inside the width of RBF ϕ_i .

- The overlapping of the RBF ϕ_i and the other RBFs is quantified by using the parameter o_i :

$$o_i = \sum_{j=1}^m o_{ij} \tag{3}$$

$$o_{ij} = \begin{cases} (1 - \|\phi_i - \phi_j\|/d_i) & \text{if } \|\phi_i - \phi_j\| < d_i \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where o_{ij} measures the overlapping of the RBF ϕ_i y ϕ_j $j = 1 \dots m$.

Applying Operators to RBFs. In CO²RBFN four operators have been defined in order to be applied to the RBFs:

- Operator Remove: eliminates an RBF.
- Operator Random Mutation: modifies the centre and width of an RBF in a percentage below 50% of the old width.
- Operator Biased Mutation: modifies the width and all coordinates of the centre using local information of the RBF environment. The technique used follows the recommendations in [4] that are similar to those used by the LMS algorithm. The error of the patterns within the radius of the RBF, ϕ_i , are calculated.

The operators are applied to the whole population of RBFs. The probability of choosing an operator is determined by means of a Mandani-type fuzzy rule based system which represents expert knowledge about the operator application in order to obtain a simple and accurate RBFN.

The inputs of this system are the parameters a_i , e_i and o_i used to define the credit assignment of the RBF ϕ_i . These inputs are considered as the linguistic variables va_i , ve_i and vo_i . The outputs, p_{remove} , p_{rm} , p_{bm} and p_{null} , represent

the probability of applying Remove, Random Mutation, Biased Mutation and Null operators, respectively.

Introduction of New RBFs. In this step, the eliminated RBFs are substituted by new RBFs. The new RBF is located in the centre of the area with maximum error or in a randomly chosen pattern with a probability of 0.5 respectively.

The width of the new RBF will be set to the average of the RBFs in the population plus half of the minimum distance to the nearest RBF. Its weights are set to zero.

Replacement Strategy. The replacement scheme determines which new RBFs (obtained before the mutation) will be included in the new population. To do so, the role of the mutated RBF in the net is compared with the original one to determine the RBF with the best behaviour in order to include it in the population.

3 Experimental Framework

In collaboration with Poolred², an initiative of the Foundation for the Promotion and Development of the Olive and Olive Oil, located in Jaén (Spain), the time series of the monthly extra-virgin olive oil price per ton at source in Spain has been obtained (see Figure 2). Concretely, the time series contains data from the 1st month of 2002 to the 12th month of 2009.

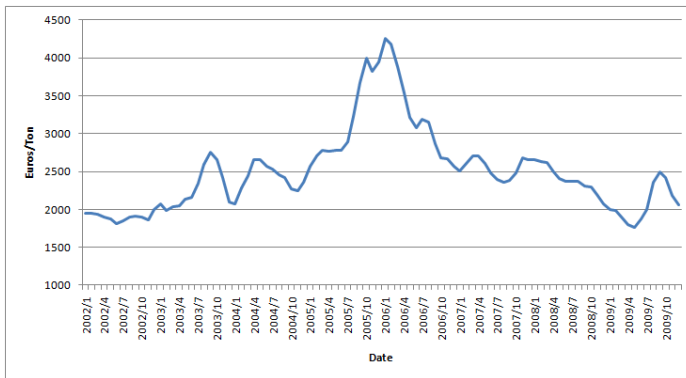


Fig. 2. Time series of the extra-virgin olive oil price

3.1 Exogenous Variables and Technical Indicators

The chosen exogenous variables or stock indexes, that contributes to predict the extra-virgin olive oil, are shown in Table 1. As can be seen the source of these

² <http://www.oliva.net/poolred/>

variables/indexes are: the cited Pooled, the Agency for the Olive Oil³ in Spain, the National Institute of Statistic of Spain⁴ and the Ministry of Industry, Tourist and Trade⁵. All these variables/indexes are monthly.

Table 1. Exogenous variables used to forecast the olive oil price

Variable	Description	Source
TgtPrice	Target Price of the extra-virgin olive oil	Pooled
OpStock	Opening stocks of olive oil	Agency for the Olive Oil
ClStock	Closing stocks of olive oil	Agency for the Olive Oil
InMarket	Trades in Internal Market of olive oil	Agency for the Olive Oil
Imports	Imports of olive oil	Agency for the Olive Oil
Exports	Exports of olive oil	Agency for the Olive Oil
ConMK	Consumption of olive oil in millions of kilos	Ministry of Industry, Tourist and Trade
GenCPI	General Consumer Price Index	National Institute
FoodCPI	Food Consumer Price Index	National Institute of Statistics

With the aim of extracting additional information of the above data, a set of technical indicators, frequently referenced in the specialized bibliography ², have been used and are shown in Table ². In this table, i_t is the value of the index at time t , H_{t-k} and L_{t-k} are the highest and lowest values respectively, during a period of time k , and H_n and L_n are the highest and lowest values respectively from the beginning of the time series.

Table 2. Technical indicators and their formulas

Feature name	Description	Formula
Momentum 1	Measures the change of an index over a time span of one moth	$i_t - i_{t-1}$
Momentum 3	Measures the change of an index over a time span of three moths	$i_t - i_{t-3}$
Momentum 6	Measures the change of an index over a time span of six moths	$i_t - i_{t-6}$
Stochastic %k	Measures the last value of the index relative to its price range over a given time period. $k = 6$ in our case.	$\frac{i_t - L_{t-k}}{H_{t-k} - L_{t-k}} \times 100$
Williams %R	Larry William's %R. A momentum indicator that measures overbought/oversold levels	$\frac{H_n - i_t}{H_n - L_n} \times 100$
Disparity6	6-day disparity. Means the distance of current price and the moving average of 6 days	$\frac{i_t}{MA_6} \times 100$

As we have managed nine exogenous variables, besides the source price of the extra-virgin olive oil itself and six technical indicators, besides the absolute or raw value of each variables, the first experiments, taking into account all the combinations, are composed by seventy input variables. All the variables and technical indicators managed can be seen in Table ⁵. Also, data are normalized in the interval $[0, 1]$.

³ <http://aao.mapa.es>

⁴ <http://www.ine.es>

⁵ <http://www.mityc.es>

3.2 Feature Selection Algorithms

In order to carry out the feature selection [7], the Weka data mining software [8] is used.

As mentioned a filter approach has been chosen because it operates independently of the learning algorithm without biasing the results, is much faster than the wrapper approach and hence can be applied to large data sets containing many features.

In Weka for feature selection tasks the feature evaluator and a search method, that defines the set of attributes, can be chosen independently. In this work, as feature evaluator the CfsSubsetEval [9] method has been chosen. CfsSubsetEval evaluates the worth of a feature subset by calculating feature-class and feature-feature correlations. Feature subsets with high correlation with the class and low intercorrelations among the features, are preferred.

With the objective of determining the best attribute subset, the following search methods, implemented in Weka, have been chosen: BestFirst [6], GeneticSearch [5], GreedyStepwise [6], LinearForwardSelection [6], ScatterSearch [3], SubsetSizeForwardSelection [6].

CfsSubEval has been run as evaluator method for all the search methods obtaining six feature selection methods.

4 Experimentation and Results

In this study the data is partitioned as is shown in the Table 3. In order to estimate prediction capacity, the Mean Absolute Percentage Error, MAPE, is calculated.

$$MAPE = \sum_i^z (| (f(x_i) - y(x_i))/f(x_i) |)/z, \quad (5)$$

where $f(x_i)$ is the predicted output of the model, $y(x_i)$ is the desired output and z is the number of patterns in the data set.

Table 3. Data sets

Data set	Training years	Test years
Test2006	2002 2003 2004 2005	2006
Test2007	2002 2003 2004 2005 2006	2007
Test2008	2002 2003 2004 2005 2006 2007	2008
Test2009	2002 2003 2004 2005 2006 2007 2008	2009

The result obtained by CO²RBFN have been compared with those obtained by four different soft-computing methods implements in KEEL [1]: FuzzyGAP [13], MLPConjGrad [10], and RBFNLMS [14]. The main parameters used are set to the values indicated by the authors. The parameters used for CO²RBFN are: Generations of the main loop = 200 and Number of RBF's = 10.

In order to achieve our objective, to forecast the extra-virgin olive oil price at source (SrcPrice) six months ahead, the absolute value of this SrcPrice and nine exogenous variables Table 1, have been chosen and extra input variables have been obtained processing each exogenous variable with six technical indicators Table 2. To these initial data sets, with seventy input variables, CO²RBFN and other soft computing methods have been applied.

With the aim of decreasing the number of input variables and increasing the interpretability of the problem, different feature selection algorithms have been applied. So, new data sets have been built with the previously selected variables. Finally, soft computing methods have been applied to these data sets and the results are analyzed.

4.1 Results Obtained with the Whole Set of Input Variables

First, CO²RBFN and the rest of soft computing methods are applied to the data sets composed by all the input variables. The results obtained average and standard deviation for 10 repetitions according to MAPE, are shown in Table 4. As can be observed, CO²RBFN has the lowest average error, followed by RBFNLMS (the other RBFN design method). CO²RBFN has also the lowest average standard deviation that implies a good robustness.

Table 4. MAPE test with the whole set of input variables

Year	CO ² RBFN	FuzzyGap	MLPConjGrad	NUSVR	RBFNLMS
Test2006	0.2366 ± 0.0418	0.2085 ± 0.1411	0.4209 ± 0.3329	0.3603 ± 0.3105	0.1953 ± 0.1090
Test2007	0.0630 ± 0.0209	0.2179 ± 0.0990	0.3901 ± 0.2338	0.2294 ± 0.2193	0.1284 ± 0.0949
Test2008	0.0999 ± 0.0194	0.1752 ± 0.0932	0.1524 ± 0.1391	0.1006 ± 0.1115	0.1156 ± 0.0944
Test2009	0.1998 ± 0.0255	0.2245 ± 0.1470	0.2747 ± 0.2254	0.1863 ± 0.0977	0.2539 ± 0.1689
Mean	0.1498 ± 0.0269	0.2065 ± 0.1201	0.3095 ± 0.2328	0.2191 ± 0.1847	0.1733 ± 0.1168

4.2 Results of the Feature Selection Algorithms

Feature selection methods, mentioned in 3.2, have been applied to the four data sets of Table 3. The results of applying feature selection methods are shown in Table 5. In this table, the first row shows the different indexes, the first column contains the different technical indicators (where Absolute means a raw variable when no technical indicator is applied) and each cell represents the number of times that an input variable (defined by the combination of row/column) is chosen by any feature selection algorithm in any data set (year). For example, the cell (row = 2 / column = 2) shows that the input variable Absolute/SrcPrice is chosen 12 times by different feature selection algorithms and data sets, but no feature selection algorithm has chosen the input variable Momentum1/SrcPrice for any year.

Thus, we can identify important exogenous variables that often are selected regardless of the technical indicator used to preprocess it. These variables, that

Table 5. Results of applying feature selection methods

	SrcPrice	TgetPrice	OpStock	ClStock	InMarket	ConMK	Imports	Exports	GenCPI	FoodCPI	Total
Absolute	12	6	0	0	0	0	0	2	0	0	20
Momentum1	0	0	0	0	0	0	0	0	5	4	9
Momentum3	11	4	0	0	0	5	0	0	0	0	20
Momentum6	23	4	0	0	0	12	0	0	12	12	63
Stochastic %k	0	12	5	1	0	0	0	0	0	1	19
Williams %R	14	6	0	0	0	4	0	2	18	10	54
Disparity6	1	0	0	0	0	0	2	0	0	13	16
Total	61	32	5	1	0	21	2	4	35	40	-

can be said that influence the price of the extra-virgin olive oil price six months ahead, are (sorted by the number of times that they have been selected): SrcPrice, FoodCPI, GenCPI, TgetPrice and ConMK. We can conclude that the SrcPrice is the most important variable to take into account in order to predict the future price of extra-virgin olive oil. There is a second group, that have been selected in a similar number of times, to predict the extra-virgin olive oil price: FoodCPI, GenCPI an TgetCPI. The last variable to highlight is ConMK that have been selected moderately. The rest of the variables are punctually selected.

In order to build the new data sets according to the results of the feature selection algorithms, the selected input variables are those that have been chosen at least one time for any feature selection algorithm in any year. This selection aims to minimize the amount of information loss.

4.3 Results Obtained with the Selected Set of Input Variables

Finally, CO²RBFN and the rest of soft computing methods are applied to data sets composed only by the selected set of input variables. The results obtained, average and standard deviation for 10 repetitions according to the MAPE error, are shown in Table 6. Also in this case, the CO²RBFN approach achieves the better result in test (in average and standard deviation) among all the algorithms compared in this study.

The results obtained, but not better, are similar to the results with the whole set of input variables. In any case the objectives of simplifying the problem and identifying for the sector the input variables that influence the future price of the olive oil have been achieved.

As conclusions, CO²RBFN has achieved the best results in average and standard deviation for the experimentations carried out. Methods based on RBFNs have maintained the error in the predictions for the data sets composed by selected variables with respect to the data sets composed by all the input variables. The rest of the methods has obtained worst results. These facts validate the use of RBFNs in forecasting problems.

Table 6. MAPE test with the selected set of input variables

Year	CO ² RBFN	FuzzyGap	MLPConjGrad	NUSVR	RBFNLMS
Test2006	0.2454 ± 0.0483	0.2419 ± 0.1579	0.4893 ± 0.3916	0.3853 ± 0.2499	0.1434 ± 0.1101
Test2007	0.0711 ± 0.0312	0.2466 ± 0.1317	1.0056 ± 0.4911	0.4040 ± 0.1359	0.1875 ± 0.0453
Test2008	0.0963 ± 0.0138	0.2381 ± 0.1061	0.5144 ± 0.3724	0.2273 ± 0.1618	0.1319 ± 0.1053
Test2009	0.2090 ± 0.0287	0.1422 ± 0.0976	0.3309 ± 0.2046	0.1902 ± 0.1578	0.2326 ± 0.1604
Mean	0.1555 ± 0.0305	0.2172 ± 0.1233	0.5850 ± 0.3649	0.3017 ± 0.1764	0.1739 ± 0.1053

5 Conclusions

In this paper a summary of the research in predicting the extra-virgin olive oil price six months ahead, carry out by SIMIDAT group, has been presented.

Authors have been contacting whit the agents involved in this sector, concretely whit the Official Market for the negotiation of futures contracts for olive oil (MFAO) and the Foundation for the Promotion and Development of the Olive and Olive Oil (Poolred).

Soft computing methods, and particularly RBFNs, have demonstrated their efficiency in the resolution of forecasting problems. For this reason authors propose CO²RBFN, a hybrid evolutionary cooperative-competitive algorithm for RBFN design in order to solve the given problem.

Different exogenous variables and technical indicators have been used, and CO²RBFN and other soft computing methods have been applied to the initial data sets. The results obtained show that CO²RBFN is the best method in measures as the average, the standard deviation.

In order to reduce the number of input variables and to increase the knowledge about the problem, different feature selection algorithms have been applied. From these results we can conclude that variables as price at source, price at destination, CPI general, food CPI and consumption influence the future price of the extra-virgin olive oil.

Finally, new data sets have been built with the previously selected variables and soft computing methods have been applied. Also for this case, CO²RBFN is the best method in measures as the average, the standard deviation.

As future work, wrapper mechanisms of feature selection will be introduced in CO²RBFN. In this way, we can observe the sets of selected variables obtained and the efficiency of the new proposal.

Acknowledgments. Supported by the Spanish Ministry of Science and Technology under the Project TIN2008-06681-C06-02, FEDER funds, the Andalusian Research Plan TIC-3928 and the Project of the U. of Jaén UJA-08-16-30.

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. *J. of Mult.-Valued Logic & Soft Computing* 17, 255–287 (2011)
2. Atsalakis, G.S., Valavanis, K.P.: Surveying stock market forecasting techniques - part ii: Soft computing methods. *Expert Systems with Applications* 36(3,Part 2), 5932–5941 (2009)
3. García, F., García, M., Melián, B., Moreno, J.A., Marcos, J.: Solving feature subset selection problem by a parallel scatter search. *European Journal of Operational Research* 169, 477–489 (2006)
4. Ghost, J., Deuser, L., Beck, S.: A neural network based hybrid system for detection, characterization and classification of short-duration oceanic signals. *IEEE Journal of Ocean Engineering* 17(4), 351–363 (1992)
5. Goldberg, D.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (1989)
6. Gütlein, M., Frank, E., Hall, M., Karwath, A.: Large scale attribute selection using wrappers. In: *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining* (1999)
7. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *The Journal of Machine Learning Research* 3, 1157–1182 (2003)
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. *SIGKDD Explorations* 11(1) (2009)
9. Hall, M., Smith, L.A.: Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In: *Twelfth International FLAIRS Conference*, MIT Press, AAAI/ (1999)
10. Moller, F.: A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 6, 525–533 (1990)
11. Pérez-Godoy, M.D., Pérez, P., Rivera, A.J., del Jesus, M.J., Carmona, C.J., Frías, M.P., Parras, M.: *co²rbfn* for short-term forecasting of the extra virgin olive oil price in the spanish market. *International Journal of Hybrid Intelligent Systems* 7(1), 75–87 (2010)
12. Rivera, A.J., Pérez-Recuerda, P., Pérez-Godoy, M.D., del Jesus, M.J., Frías, M.P., Parras, M.: A study on the medium-term forecasting using exogenous variable selection of the extra-virgin olive oil with soft computing methods. *Applied Intelligent* 34(3), 331–346 (2011)
13. Sánchez, L., Couso, I.: Fuzzy random variables-based modeling with ga-p algorithms. In: Bouchon, B., Yager, R.R., Zadeh, L. (eds.) *Information, Uncertainty and Fusion*, pp. 245–256 (2000)
14. Widrow, B., Lehr, M.A.: 30 years of adaptive neural networks: perceptron, madaline and backpropagation. *Proceedings of the IEEE* 78(9), 1415–1442 (1990)

SMS Normalization: Combining Phonetics, Morphology and Semantics

Jesús Oliva, José Ignacio Serrano,
María Dolores del Castillo, and Ángel Iglesias

Bioengineering Group, Spanish National Research Council (CSIC),
Carretera de Campo Real, km. 0,200, La Poveda, Arganda del Rey,
CP: 28500, Madrid, Spain

{jesus.oliva,jignacio.serrano,md.delcastillo,angel.iglesias}@csic.es

Abstract. The language used in electronic communications such as e-mails, chats and SMS texts presents special phenomena and important deviations from natural language. Typical machine translation approaches are difficult to adapt to SMS language due to the many irregularities this kind of language shows. This paper presents a new approach for SMS normalization that combines lexical and phonological translation techniques with disambiguation algorithms at two different levels: lexical and semantic. The results obtained by the system outperform some of the existing methods of SMS normalization despite the fact that the corpus created has some features that complicates the normalization task.

1 Introduction

SMS Language, also known as texting language, is a new way of communication developed in the last ten years as a result of the massive use of electronic communications all around the world. Only in Spain 25,000 million SMS messages are sent per year, 365,000 million were sent in 2006 only in Occidental Europe. Besides, there exists an incalculable amount of messages written with this kind of language in chat rooms and instant messaging programs. Nowadays, there is not much work about SMS normalization (see [15] for English or [8] for French) in spite of the fact that this massive use of SMS language makes suitable to develop normalization systems that help to process all this information. There exist many applications for which SMS normalization could be really useful. For example, search engines for noisy text documents such as emails or blogs, text correction over the web or text-to-speech systems. Another motivation to implement this kind of normalization systems is the preservation of the native languages. SMS language does not take into account many orthographical and grammatical rules. So the uncontrolled use of this kind of language could lead to a deterioration of the original languages. SMS language is not only a marginal and informal variant of the standard language but a process of language evolution that has got the potential to modify the standard language [3]. Finally, the methodology used in SMS language normalization could be a source of ideas to

be used in general machine translation systems since SMS normalization could be treated as a subproblem of Machine Translation.

This paper presents a method for the normalization from Spanish SMS language texts into Spanish natural language texts. The method proposed carries out the lexical and semantic disambiguation making use of a combination of lexical resources such as SMS and Spanish dictionaries or the lexical database WordNet. The main contributions of this work are not only the whole normalization system, but also a new phonetic-based distance between SMS words and Spanish words. This distance in combination with a phonetic Spanish dictionary built ad-hoc for this application enables the extraction of all the Spanish words phonetically similar to a SMS word. Our method outperforms some of the existing methods of SMS normalization despite the fact that the corpus created has some features that complicate the normalization task. For example, a higher rate of unknown words or a lower BLEU score of raw text (a well-known metric that compares the translation given by a system and the real translation) [12].

2 SMS Language Features

The main features of Spanish SMS language can be found in [7]:

- Phonetic abbreviations and vowel elimination are the phenomena mostly used in SMS language. Phonetic abbreviations are those which keep a similar phonetic structure with the real word: *txt - text*.
- Non-phonetic abbreviations are also widely used. These abbreviations have no phonetic likeness with the word they refer to. For example: *xxx - kisses*.
- Blank characters are sometimes omitted after a punctuation mark to save characters, making necessary a preprocessing step that splits the SMS sentence in its forming words. For example, in SMS language can be found: *Hi!hw r you?*
- In addition, the orthographical rules (in Spanish, particularly the accentuation rules) are often not taken into account, thereby increasing the ambiguity.

Based on the conclusions obtained by [7], two hypotheses are assumed by our method:

1. A SMS phonetic abbreviation always presents the same consonants, or consonants phonetically equivalent, as the referred Spanish word (the only exception is letter 'h', which has no phonetic transcription). This seems to be a very logical assumption because the elimination of a consonant would lead to a very different word in terms of phonetics.
2. The vowels that appear in an SMS phonetic abbreviation are always in the same order in the referred Spanish word. This is also a very logical assumption since it has no sense to add a vowel that is not in the original word or to add it in a different position.

3 SMS Translator: System Architecture

SMS normalization requires the processing of special symbols and phonetic abbreviations and the disambiguation at two main levels. In order to solve these problems, a system composed of three modules is proposed. Preprocessing, translation and disambiguation at two different levels are the processes carried out by the system. In order to show the way the proposed normalization method works, a Spanish sample sentence will be used: *Pues tiene dos teléfonos móviles* (in English: *So he has two mobile phones*) written in SMS language as: *Pues tiene 2telfs mvls*.

3.1 Preprocessing Module

One of the characteristics of SMS language is the possible absence of blanks after a punctuation mark or number because of the space limitations that SMS imposes. For example, tokens like *2telfs* or *hola.como estás?* could be found. Therefore, the first task of the preprocessing module is to do a correct tokenization of the SMS text, which implies dividing the tokens consisting of letters and signs and uppercasing the words after a dot sign. However, SMS language can present abbreviations and emoticons, composed of letters and signs, which should not be split to be translated correctly. So, before splitting a token consisting of letters and punctuation marks or numbers, the module tries to translate it by using a Spanish SMS dictionary with more than 11,000 entries, provided by the 'Asociación Española de Usuarios de Internet' ¹ (Spanish Internet Users Association). If the word is in the SMS dictionary, this module stores the possible translations, otherwise, this module tokenizes it obtaining the token (or tokens) of letters and the token (or tokens) of symbols.

In the example presented, the preprocessing module would find that the word *2telfs* is formed by numbers and letters. So, after trying to translate it using the SMS dictionary without success, the module would break the single word and it would outcome two tokens: *2* and *telfs*.

3.2 Translation Module

The translation module gets all the possible translations of the SMS words. In order to do it, the module has to deal with three kinds of words: phonetic abbreviations, which are obtained by removing some vowels of the original word like *mvls* - *móviles*, non-phonetic abbreviations like *xa* - *para*, and real words. The translation module uses an SMS dictionary to deal with non-phonetic abbreviations and a Spanish phonetic dictionary to deal with phonetic abbreviations and real words. The output of this module is the union of the lists of possible translations extracted from the SMS dictionary and the Spanish phonetic dictionary.

¹ <http://www.diccionariosms.com/contenidos/>

Table 1. Input and output of the translation module. The English translations of the different words are: puedes - can, dos - two, tú - you, teléfonos - phones, pues - so, tiene - has, móviles - mobile, muebles - furniture, amables - kind.

Original word	Pues	Tiene	2	Telfs	Mvls
SMS dictionary	Puedes	–	Dos, tú	Teléfonos	–
Spanish Phonetic dictionary	Pues	Tiene	–	–	Móviles Muebles Amables
Resulting possible translations	Puedes, Pues	Tiene	Dos, Tú	Teléfonos	Móviles, Muebles, Amables

SMS Dictionary. The translation module tries to find each word in the SMS dictionary, which stores non-phonetic abbreviations. If the word is found there, all the possible translations are stored. As stated before, a SMS word could be similar to an SMS abbreviation and to a real Spanish word. So, even if the SMS word is found in the SMS dictionary, the translation module has to check whether it is similar to a real Spanish word. Therefore, the module uses the Spanish phonetic dictionary explained in the following point.

Spanish Dictionary. A Spanish phonetic dictionary has been specifically built to make the translation of phonetic abbreviations easier. The use of such a phonetic dictionary in combination with a new similarity metric proposed further on is one of the main contributions presented in this paper. The dictionary consists of a special code that reflects the phonetic uses for each consonant in Spanish SMS language and a string storing the vowels and its positions in the word. This allows to map the SMS word to several natural language words that share the same consonant base. The possible rewrites are ranked by a weighted Levenshtein distance which is thresholded to select admissible rewrites. The consonant code groups the consonants which can be used indistinctly in Spanish SMS language because of their phonetic similarity (such as *B* and *V*). These phonetic groups are: *B-V*, *C-Q-K*, *G-J-W*, *R-'RR'*, *X-'CH'* and *Y-'LL'*. Each entry of the dictionary is formed by a consonant string, composed of the codes of each consonant, and a vowel string composed of the vowels and their order in the Spanish word.

The translation module tries to find out whether the SMS word is a phonetic abbreviation of a real word or not. To do this, we assume the two hypotheses pointed in section 2. According to these two assumptions, the translation process works as follows: given a SMS word, first of all the word is searched in the Spanish phonetic dictionary to get the entries that have the same coded consonant string. Then, for each of the entries, the translation module computes the similarity of the associated vowels strings.

The similarity between strings is computed by using a new metric proposed here as a modification of the Levenshtein distance [9]. Levenshtein distance gives a cost to the operations of insertion, elimination or substitution of characters. Our measure slightly penalizes the insertion of a vowel. However, the substitution of a vowel has a high cost unless it is substituted by the same vowel with an accent. The elimination operation is very highly penalized since SMS language does not imply the insertion of characters. The cost values for each operation and the threshold have been fixed empirically using a subset of 20 messages not used in the evaluation.

The output of the translation module would be a list of sets of words in which each set represents the possible translations of the corresponding word in the original sentence. Following our example, we would get the results shown in Table 11.

3.3 Disambiguation Module

Lexical Disambiguation. In order to do the lexical disambiguation task, the system uses the open source suite of language analyzers "Freeling 2.1" [2] [4] with some modifications to deal with our particular problem. At first, the input of this module is a sentence that in each position does not have a single word but a list of possible words. Each list is composed by all the possible translations for each word received from the translation module. For the list associated to each position in the sentence, the module stores all the possible POS tags using the dictionary provided by Freeling and also uses Freeling to determine the most probable combination of these POS tags. Once Freeling obtains the selected POS tag for each position, the disambiguation module selects the words of the corresponding list that have that POS tag among its possible POS tags.

The process followed by this submodule can be seen in Table 2. The output of this submodule is formed by the words whose POS tag is the same as the selected POS tag given by Freeling. Note that the lexical disambiguation process has two positive effects: at the very best, lexical disambiguation is enough to completely eliminate ambiguity (see first and third words) and also in those cases that ambiguity still remains (see last word), the number of options is reduced, making easier the semantic disambiguation task.

Semantic Disambiguation. The semantic disambiguation module uses WordNet 2.1 [3] [6], which due to its hierarchical structure and the existence of term definitions in each sense, enables the design and use of many kinds of semantic similarity measures between words. In order to access WordNet we used JWordNet-Similarity [4], a Java interface that implements a variety of semantic similarity and relatedness measures including the one used by the system proposed here, the Extended Gloss Overlap Measure [2]. Besides, given that

² <http://garraf.epsevg.upc.es/freeling/>

³ <http://wordnet.princeton.edu/>

⁴ <http://www.eml-research.de/english/research/nlp/download/jwordnetsimilarity.php>

Table 2. Lexical disambiguation process. The subscripts show the possible tags for each word. The English translations of the different words are: puedes - can, dos - two, tú - you, teléfonos - phones, pues - so, tiene - has, móviles - mobile, muebles - furniture, amables - kind. The abbreviations mean: V - verb, SC - subordinate conjunction, DT - determiner, PN - pronoun, N - noun and ADJ - adjective.

Possible Translations	<i>Puedes_V</i>	<i>Tiene_V</i>	<i>Dos_{DT}</i>	<i>Teléfonos_N</i>	<i>Móviles_{ADJ,N}</i>
	<i>Pues_{SC}</i>		<i>Tú_{PN}</i>		<i>Muebles_N</i>
					<i>Amables_{ADJ}</i>
Possible tags	V	V	DT	N	ADJ
	SC		PN		N
Selected tag	SC	V	DT	N	ADJ
Output	Pues	Tiene	Dos	Teléfonos	Móviles Amables

WordNet is an English ontology, we used the Spanish WordNet-based semantic information provided by FreeLing to construct an English-Spanish dictionary based on semantics. FreeLing provides for each Spanish word its corresponding English WordNet synsets (extracted from EuroWordNet⁵), so the English-Spanish dictionary was made up by inversely translating the synsets for its associated words in WordNet.

The algorithm used by the system to deal with the semantic ambiguity is a novel adaptation of the Maximum Relatedness Disambiguation (MRD) algorithm [13] to make the disambiguation among possible translations instead of among possible senses. The adaptation presented here takes the words with the most related senses (from now on we will call a sense-pair each pair of senses formed by a sense of the target word and a sense of one of the words in the context window) among the target word and the words in it context.

The algorithm processes the input sentence from left to right. When the first ambiguous word (target word) is found, the context window is built. This window is formed by the words placed just before and after the target word whose type is present in WordNet (that only takes into account nouns, verbs, adjectives or adverbs). The window size used in our system was 3, which included the target word and one word to its left and right. At his point, we followed the claim done by [10] regarding that words farther away from the target word are less likely to be related to words close to the target word. Once the context window is built, the algorithm takes each possible translation of the target word and computes the relatedness of all its senses with all the senses of all the words in the context window. Note that words to the left in the context window cannot be ambiguous either because they are not ambiguous or because they have already been disambiguated. However, words to the right could be ambiguous, so the algorithm has to compute the relatedness with all the possible words. For each

⁵ <http://www.illc.uva.nl/EuroWordNet/>

Table 3. Semantic similarity of sense-pairs of words *Teléfono* and *Amable* and *Teléfono* and *Móvil*

		TELÉFONO			
		Phone	Telephone	Telephony	Number
	Gracious	0.75	1.38	1.0	0.41
AMABLE	Good-hearted	1.33	2.0	1.0	0.91
	Kind	0.67	0.83	0.33	0.73
MÓVIL	Mobile	2.89	2.83	1.67	0.97

word in the context window, the algorithm gets the most related sense-pair and computes the score of each of the possible translations of the target word as the sum of the scores of the most related sense-pairs. Finally, the algorithm selects the option with the highest score. Table 3 shows the comparisons done by the algorithm in order to disambiguate the last word of the example sentence. The context window is made by the word *telephone* which has 4 senses in WordNet. The most similar sense-pair is the one made by the sense *phone-mobile*. Thus, *móviles (mobile)* is selected.

4 Evaluation

4.1 Data Set

There are not many studies about SMS normalization, particularly in Spanish, so there is not a suitable benchmark data set for the evaluation of Spanish SMS normalization systems. Thus, a data set was built with messages written by undergraduate students of the University of Extremadura to the magazine “Extremadura Universa”⁶. The SMS collected were sent to the magazine between January 2003 and March 2008, and all of them are anonymous. Some important features of the corpus are the following: unknown tokens (those which do not correspond to a Spanish word) account for a 78.71% of the total number of tokens while ambiguous tokens (those which have more than one possible translation) reach a 37.96%. The number of ambiguous tokens is an important measure that complements the number of unknown tokens. If the number of unknown tokens is very high but they have only one possible translation, a simple dictionary-based method could obtain great results. Also BLEU 3-gram score of raw text is given to compare the a-priori difficulty of the task. In our corpus the BLEU score of raw text is 0.1243. It is important to note that the corpus used to evaluate our method seems to have some features that make the task more complicated than the one faced in other related papers. The corpus used by [8] presents only a 32.7% of unknown words which is much lower than the 78.71% of our corpus, making easier the normalization task. Also [1] uses a corpus with an initial BLEU

⁶ <http://www.elperiodicoextremadura.com/suplementos/universa/>

Table 4. Number of words processed, errors, word error rate and BLEU score of each module and the whole system

Module	Total Words	Errors	WER	BLEU score (3-gram)
Preprocessing + Translation	1557	112	0.071	0.469
Lexical Disambiguation	665	55	0.083	0.624
Semantic Disambiguation	276	48	0.174	0.805
OVERALL	1557	215	0.138	0.805
Baseline	1557	1009	0.648	0.469

3-gram score of 0.5784, which shows that the source messages are much more similar to the real translations than the ones used in this paper.

4.2 Evaluation Process

Not only the evaluation of the whole system has been carried out. Also, some stages of the algorithm were evaluated separately to observe the weaknesses and strengths of each of the steps of the system. The evaluation process works as follows: first of all, when each of the modules finishes its work, the output (i.e., the list of possible translations for each word) is stored. When the translation is done, for each word, if the real word is not included in the list of possible translations given by the module, the number of errors of the corresponding module is increased. When an error is detected on a module, the translation process continues. But, in order to compute the errors of each module, we don't take into account the errors made by the previous modules. To obtain the BLEU scores shown in Table 4, we select the first option of the possible translation lists generated by each module. To evaluate the method proposed we used the word error rate. Also BLEU metric is given to allow comparisons with other similar studies like the ones proposed by [18,11]. In addition to word error rate, we used the sentence error rate, which provides information about the distribution of errors among sentences.

4.3 Results and Discussion

The results obtained by each of the modules of the normalization system proposed, and the overall precision of the method are shown in Table 4. As a baseline experiment we consider a simple dictionary-based system. The baseline system preprocesses and translates the input words in the same way as the preprocessing and translation modules presented in this paper. However, for obtaining the final normalization of each message, the baseline system takes the first of the words in the possible translations list of each token.

The sentence error rate obtained by the system is 0.609, i.e., 56 normalized sentences out of the total 92 sentences contain an error. This value is much better

than the 0.75 reported by [11] and is similar to the approximate 0.60 reported by [8]. Also Table 4 shows good and promising results of the normalization system proposed here. The global BLEU score of 0.8054 is similar to the 0.8070 reported by [1] for English, and the approximately 0.8 reported by [8] for French. Furthermore, the system highly outperforms the method proposed by [11] which obtains a BLEU score of 0.68. It is important to note that these results are obtained with a corpus with some features, such as a higher rate of unknown words or a lower BLEU score of raw text, that complicate the normalization task.

Errors in preprocessing and translation are mainly due to the words that are not contained in the dictionaries, such as diminutives, nicknames or proper nouns, which are commonly used in SMS language. The problem with these kinds of words could be eliminated by using dictionaries that include diminutive forms or by using some entity recognition method capable of detecting these forms. Also it would be interesting to test some speech recognition approaches that provide multiple segmentation candidates attending to phonetics.

Lexical disambiguation obtains great results. Observing the errors done by this module, it seems that this is not a critical point to improve. Errors are mainly detected in the disambiguation of words which have some previous words wrongly translated, leading the disambiguation to fail.

The main source of errors in the semantic disambiguation is only present in languages, like Spanish, that present conjugated verbs or gender-marked nouns. In fact, in many cases it is not possible to obtain the correct form of a verb or the correct gender of a noun because there is no information in the message to choose the correct form. The total amount of errors caused by a bad conjugation of a verb is 14, which supposes a 29.17% of the errors made by this module and a 6.51% of the total errors made by the system. Regarding gender, the errors caused by this problem suppose the 35.42% of the errors made by the semantic disambiguation module and a 7.91% of the overall errors of the system. It is important to remark that it is impossible to correct some of these errors because the election of the correct form depends on external circumstances that are not contained in the message (for example, the gender of the receiver of the message). Also, it is important to note that these problems are only present in certain languages such as Spanish or French, but not in English. So this fact should be taken into account in the comparison of this method with methods designed to work with English.

5 Conclusions

This paper presented a novel approach to SMS normalization. The system is based on a three-module architecture built up by a preprocessing module, a translation module and a disambiguation module. The preprocessing module divides the words correctly, taking into account some special characteristics of SMS language such as the possible absence of blanks. The translation module tries to obtain all the possible translations of each word making use of two different dictionaries to deal with phonetic and non-phonetic abbreviations. Finally,

the disambiguation module chooses the correct translation for each word among all the possible translations given by the translation module. In order to make this choice, the disambiguation module carries out a process of lexical disambiguation and a process of semantic disambiguation, taking into account context information and the semantic knowledge stored in WordNet. The main contributions of this work are not only the whole normalization system, but also a new phonetic-based distance between SMS words and Spanish words based on a weighted Levenshtein distance. This distance in combination with a phonetic Spanish dictionary built ad-hoc for this application enables the extraction of all the Spanish words phonetically similar to a SMS word. The proposed system outperforms some of the existing methods of SMS normalization despite the fact that the corpus created has some special features, such as a higher rate of unknown words or a lower BLEU score of raw text, that complicate the normalization task.

References

1. Aw, A., Zhang, M., Xiao, J., Su, J.: A phrase-based statistical model for sms text normalization. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions, pp. 33–40. Association for Computational Linguistics, Morristown (2006)
2. Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness. In: Proceedings of the Eighteenth International Joint conference on Artificial Intelligence, Acapulco, pp. 805–810 (2003)
3. Baron, N.: Computer mediated communication as a force in language change. *Vis. Lang.*, 118–141 (2004)
4. Carreras, X., Chao, I., Padró, L., Padró, M.: Freeling: An open-source suite of language analyzers. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, LREC 2004 (2004)
5. Choudhury, M., Saraf, R., Jain, V., Mukherjee, A., Sarkar, S., Basu, A.: Investigation and modeling of the structure of texting language. *Int. J. Doc. Anal. Cognit.* 10(3), 157–174 (2007)
6. Fellbaum, C.: *WordNet: An Electronic Lexical Database*. MIT Press (1998)
7. Ferri, S.: El fenómeno de economía lingüística en el lenguaje SMS: breve estudio experimental en alumnos de 16 años, pp. 255–270 (2005)
8. Kobus, C., Yvon, F., Damnati, G.: Normalizing sms: are two metaphors better than one? In: COLING 2008 (2008)
9. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. *Tech. Rep.* 8 (1966)
10. Michelizzi, J.: Semantic relatedness applied to all word sense disambiguation. Master's thesis, University of Minnesota (2005)
11. Guimier de Neef, É., Debeurme, A., Park, J.: Tilt correcteur de sms: évaluation et bilan qualitatif. *Actes de TALN 2007* (2007)
12. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. *Tech. Rep.* RC22176 (W0109-022), IBM Research Division (2001)
13. Pedersen, T., Banerjee, S., Patwardhan, S.: Maximizing semantic relatedness to perform word sense disambiguation (2005)

Multiscale Extension of the Gravitational Approach to Edge Detection

Carlos Lopez-Molina^{1,2}, Bernard De Baets², Humberto Bustince¹,
Eduarne Barrenechea¹, and Mikel Galar¹

¹ Dpto. Automatica y Computacion, Universidad Publica de Navarra,
Pamplona, Spain

² Dept. of Applied Mathematics, Biometrics and Process Control,
Ghent University, Gent, Belgium

Abstract. The multiscale techniques for edge detection aim to combine the advantages of small and large scale methods, usually by blending their results. In this work we introduce a method for the multiscale extension of the Gravitational Edge Detector based on a t-norm T . We smoothen the image with a Gaussian filter at different scales then perform inter-scale edge tracking. Results are included illustrating the improvements resulting from the application of the multiscale approach in both a quantitative and a qualitative way.

1 Introduction

In the literature there exist a wide amount (and diversity) of edge detection methods, featuring very diverse techniques. The inspirations for such techniques come from different fields, including soft computing, physics or statistics. Nevertheless, at some point of the processing, most of them evaluate the intensity or color of the pixels in the neighbourhood of each pixel. This evaluation is performed in many different ways, such as local measurements, discrete convolutions or pattern-matching.

When a neighbourhood-based evaluation is to be performed, it is necessary to define the size it should have. That is, how many pixels are to be considered as neighbours of each pixel. Some edge detection methods make use of fixed-size neighbourhoods (as FIRE [23] or the convolution with the Sobel [25] or Prewitt [20] operators), while others adapt it based upon the values of their parameters (as the LOG [15] or Canny [3] operators). Even if some operators are meant to be infinite, they are always implemented as a discrete filter with finite support. Generally, smaller scales are related to spatially accurate edge detection, but also with higher sensitivity to noise. In the case of some specific detectors, the relationship between both of them has been studied. The most relevant case is the Canny method. Canny [3] grounds its development in the modeling and optimization of three criteria, being two of them the *spatial accuracy (localization)* and the *single response to an edge*. As one of the conclusions, Canny stated that there was necessarily a trade-off between the accuracy of the

response and the ability of missing spurious responses. This work has been later revisited by different authors as Demigny [6] or McIlhagga [17]. However, it is accepted the fact that larger scales make the detectors more robust against noise, textures and spurious edges, to the cost of potentially displacing them from their true position [19,12]. Some authors have aimed to determine the better-suited scale for a detector, but no consensus has been reached so far [8].

In this work we elaborate on an edge detection method based on fixed 3×3 neighbourhoods, extending it with notions from multiscale theory. More specifically, we perform edge detection on increasingly smoothed versions of the image and the combination of their results.

In Section 2 we analyze the scaling problem of the gravitational approach to edge detection, then introduce a multiscale algorithm. Section 3 includes some quantitative experiments, while some conclusions are drawn in Section 4

2 The Multiscale Gravitational Edge Detector Based on a t-norm T

2.1 The Gravitational Approach

In the original *gravitational approach* [26] to edge detection, each pixel in the image is taken as a body of mass equal to its intensity. Then, the position of the pixel is associated a gradient equal to the sum of the gravitational forces its immediate neighbors produce on it. Considering the situation depicted in Fig. 1, we have $\mathbf{g}_{i,j} = \sum_{k=1,\dots,8} \mathbf{F}_k$.

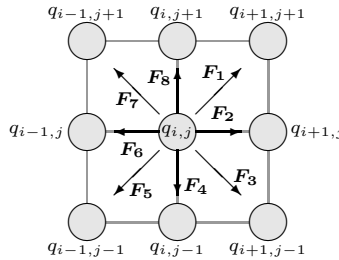


Fig. 1. Gravitational forces acting on a given pixel

An extension of the gravitational approach, namely Gravitational Edge Detector based on a t-norm T (GED- T), was introduced in [13], allowing the substitution of the product of the masses (in the calculation of the gravitational forces) by any other t-norm [13]. The effect and usability of each t-norm was studied. For example, in Figure 1, the force \mathbf{F}_1 is

$$\mathbf{F}_1 = \frac{T(q_{i,j}, q_{i+1,j+1})}{|\mathbf{r}|^2} \cdot \frac{\mathbf{r}}{|\mathbf{r}|} \tag{1}$$

¹ A triangular norm (t-norm) is a mapping $[0, 1]^2 \rightarrow [0, 1]$ that is increasing, commutative, associative and has neutral element 1.

where \mathbf{r} stands for the vector connecting the pixels (in this case $q_{i,j}$ and $q_{i+1,j+1}$).

The GED- T was shown to be competitive with the Canny method, the reference in the field. Even if performing worst in an average scenario, it outperformed the Canny and Sobel methods in a significant amount of natural images [13].

2.2 Multiscale Edge Detection

As explained in Section 1 there is no easy solution for the scale-determination problem. In fact, Torre and Poggio mention that, in order to characterize all of the possible intensity changes, *derivatives of different types, and possible different scales* would be needed [27]. Lindeberg [12] present a relevant study of the behaviour of the edges with respect to the amount of smoothing the image, aiming the automatic scale determination. The Anisotropic Diffusion, in the sense of Perona and Malik [19], is also related to this idea, since it aims to combine the small-scale filtering on the edges with the larger scale filtering of the objects surface. An alternative direction is, instead of choosing the best possible scale, combining the results obtained with many of them. This idea is based on the fact that *any feature at coarse level of resolution is required to possess a 'cause' at a finer level of resolution, although the reverse is not true* [19]. As pointed out by Konishi *et al.* [10], this implies that edges existing at coarse scales continue to exist at small scales. That is, we assume that the actual edges should appear at any possible scale, while the noise and spurious responses should disappear at larger ones. Hence, we only need to find a way to combine the spatial accuracy of the detection at the small scales with the reliability of the classification at the larger scales. Of course, we have to manage the fact that edges at larger scales do not necessarily correspond (spatially) to their positions at the smaller scales. The scale factor in edge detection has received some attention from the community in the last 20 years [14], and many authors have further developed multi-scale methods [21,10,24,5].

2.3 Multiscale Evolution of the GED-T

The GED- T has problems for scaling the neighbourhood of masses in Fig. 1, mainly due to the nature of the intensity changes measurement. Pixels outside the 3×3 windows may be considered, but their influence decreases drastically, since the force they induce on the central pixel is inversely proportional to the squared distance to the central pixel. Hence, larger windows tend to raise the computational cost (due to the larger amount of forces calculated), but produce similar results. This feature of the GED- T limits its ability to produce good results, especially in high-noise environments.

Since it is not worth scaling the neighbourhood size, we propose to smooth the original image with different Gaussian filters, and combine the results obtained thereafter. That is, to vary the scale of the Gaussian filter used in the preprocessing stage. Filters produced with large values of σ tend to oversmooth images, but they are very effective suppressing noise in the image and allow the detection of qualitatively new edges in the image (see [12] for some examples).

This approach is similar to that of, for example, Qian and Zhong [21] for extending the LOG method [15]. The idea of continuously smoothing a signal (in this case, an image) has been widely studied in the literature. Specially interesting is the case of the Gaussian smoothing, which is referred as Gaussian scale-space (GSS). Different studies on the GSS (introduced by Iijima [28], but popularized after Witkin [29]) have been presented by Babaud *et al.* [1], Yuille and Poggio [30] and Florack and Kuijper [7], among others. The developments on the study of the GSS gave rise to its application to several tasks, such as filtering based on mathematical morphology [9], histogram analysis [4] or clustering [11].

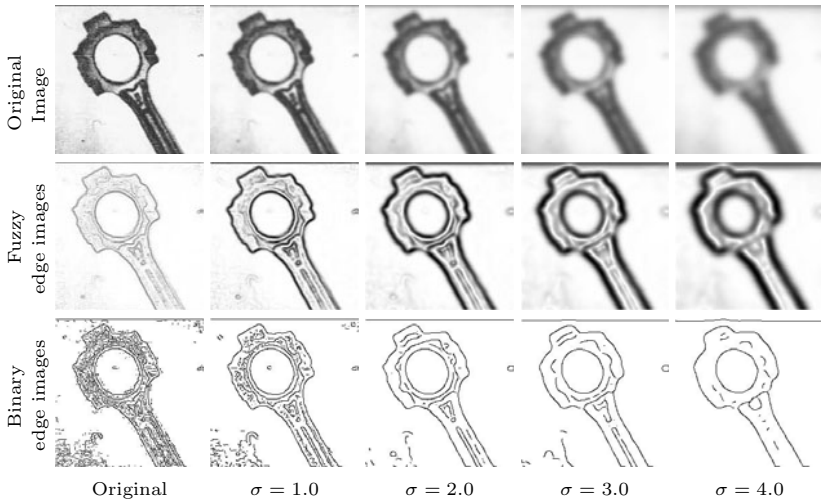


Fig. 2. Edge images generated on the span image applying the GED- T_M after Gaussian smoothing with different σ . Fuzzy edge images are converted into binary edge images using NMS and the Rosin method for thresholding.

Examples of edge images obtained with GED- T_M on images smoothed with different amount of Gaussian smoothing are included in Fig. 2. These edge images have been binarized after the GED- T procedure. In Fig. 2 we observe how larger values of σ lead to better-looking edges, and to an almost complete removal of the spurious responses. The spurious responses due to high-frequency signals (as the texture of the span) disappear with relatively low σ , while the lower frequency ones (as the imperfection on the right side of the image) only disappear with the largest of the values of σ . However, it is also noteworthy how the shape of the span is progressively degraded.

Let G_σ be the Gaussian convolution operator. Given a set of values of the standard deviation of the Gaussian convolution, $\Omega = \{\sigma_1, \dots, \sigma_n\}$, we generate n versions of the image I . The GSS is therefore sampled with n images $\mathbb{I} = \{G_{\sigma_1}(I), \dots, G_{\sigma_n}(I)\}$. We apply the GED- T on each of the images, then threshold the fuzzy edge image using non-maxima suppression [3] and the Rosin

method for thresholding [22]. That is, we generate a sequence of n fuzzy edge images $Z_i = \text{GED-}T(G_{\sigma_1}(I))$, later turned into binary images B_i . Once we have constructed the set of edge images $\mathbb{B} = \{B_1, \dots, B_n\}$, we consider that the position (i, j) is an edge pixel if:

- (C1). It is an edge at the finest scale, *i.e.* $B_1(i, j) = 1$ and
- (C2). It is displaced at most T_d positions in two consecutive edge images B_i, B_{i+1} .

The constraint C1 is very easy to test. However, in the validation of C2 we have to track the position of the edge pixel at each B_i . In order to do so, we increasingly check the position of the edge pixel. We assume that the position of the edge at the next step (if any) is the closest edge point at that scale. In case the closest edge point is further away than T_d positions, then we assume the response is due to another edge, and discard the position (i, j) . In this way, we aim to combine the ability to remove spurious edges of the large values of σ (C2) with the spatial accuracy of the small ones (C1). The procedure to do the tracking is included in Algorithm 1. This procedure tracks the edges from finer to coarser scale, in opposition to other works using coarse-to-fine tracking [10].

Data: A set of images $\mathbb{B} = \{B_1, \dots, B_n\}$, a distance threshold T_d

Result: A binary edge image B

```

begin
  for every edge pixel  $(i, j)$  of  $B_1$  do
     $s = 1;$ 
     $(i, j)' = (i, j);$ 
     $\delta = 0;$ 
    while  $s < n$  and  $T_d \geq \delta$  do
      // Update the position of the edge
       $\delta = \min(d((i, j)', (k, l)) \mid B_{(s+1)}(k, l) = 1);$ 
       $(i, j)' = \operatorname{argmin}_{(k, l)}(d((i, j)', (k, l)) \mid B_{(s+1)}(k, l) = 1);$ 
      // Update edge image
       $s = s + 1;$ 
    end
    if  $s \geq n$  then
       $B(i, j) = 1;$ 
    else
      Discard  $(i, j);$ 
    end
  end
end

```

Algorithm 1: Procedure for fine-to-coarse edge tracking.

We refer to our proposal as multiscale extension of the GED- T , briefly MGED- T . The overall processing of the MGED- T is as presented in Fig. 3. As an example of the performance of the procedure, we have applied the MGED- T on the span

image used in Fig. 2 with three different t-norms. The selected t-norms are the product (T_P), the minimum (T_M) and the Lukasiewicz t-norm (T_L) [13]. We have used 4 sets of values of σ , the Euclidean distance d and $T_d = 1.5$ (it includes all the pixels in a 3×3 window centered at the pixel). As illustrated in Fig. 4, most of the noise and spurious responses are removed, but the silhouette of the span is still placed in the actual intersection of the objects.

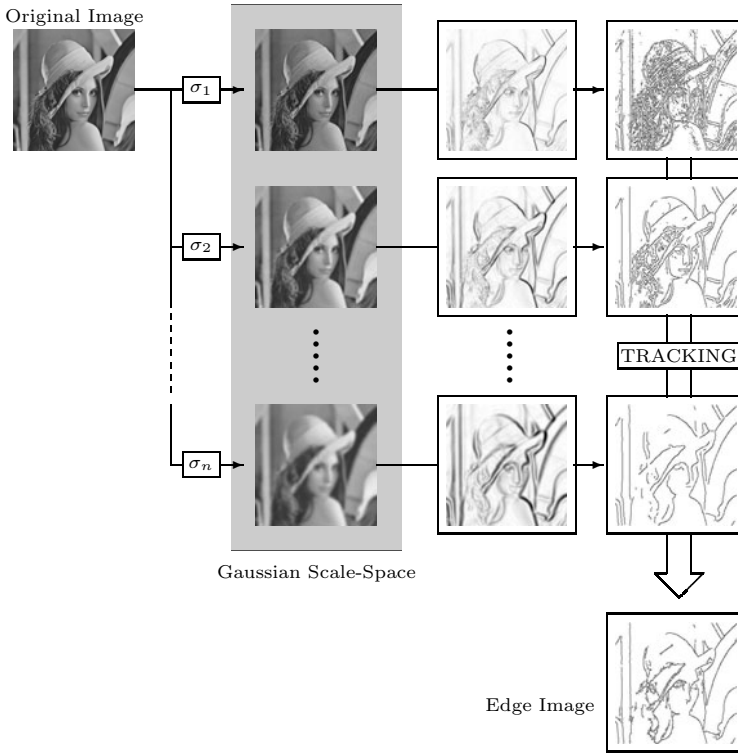


Fig. 3. Schematic visualization of the Multiscale GED- T

Since the algorithm elaborates on (and aims to improve the performance of) the GED- T , it is necessary to estimate the computational overhead it implies. We assume that the cost of the GED- T is $\mathcal{O}(M \cdot N)$, where M and N are the number of rows and columns of the image, respectively. The cost of MGED- T having n different values of σ , is $\mathcal{O}(n \cdot M \cdot N + |B_1| \cdot n)$. That is, the cost of performing n times the GED- T procedure and then tracking the points of B_1 along (up to) n edge images. We can safely assume that most of the points in the image do not belong to any edge, and hence $|B_1| \ll M \cdot N$. Therefore, $\mathcal{O}(n \cdot M \cdot N + |B_1| \cdot n) \approx \mathcal{O}(n \cdot M \cdot N)$, that is, n times the computational cost of the original GED- T .

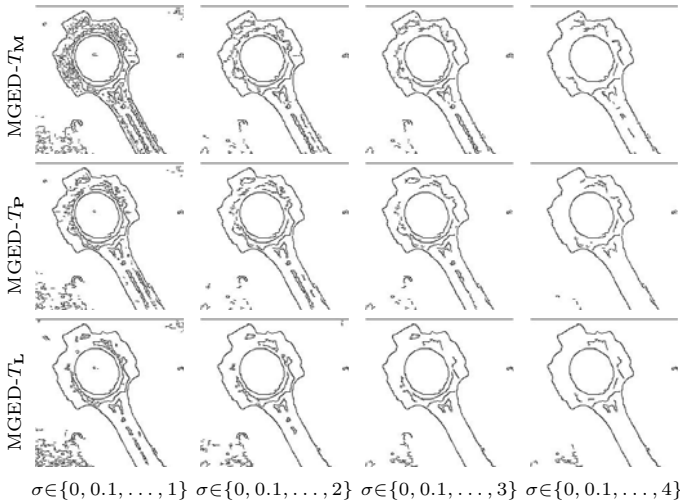


Fig. 4. Edge images generated on the span image applying the MGED- T_M , the MGED- T_P and the MGED- T_L using different sets of values of σ

3 Some Practical Experiments

In [13] the GED- T was shown to be competitive with the Canny method, the reference method in the field. It performed worst than the Canny method in average, but beat it in a significant number of cases. The multiscale edge detection methods are meant to maintain the amount of true positive (TP) responses while decreasing the number of false positives (FP). In this experiment we attempt to compare the evolution of both statistical features (TP and FP) when using different values of σ in the Gaussian smoothing. That is, whether the extra overhead the multiscale version implies is worth it or not. Note that the performance measures based on statistical features are not completely satisfactory, since they do not consider the overall shapes in the edge image. Moreover, they penalize in the same way pixels being at very different distances of the true edges [2, 18]. However, even if they are not useful for evaluating the overall quality of an image, they illustrate the specific fact we want to investigate in this experiment.

For testing we select the first 50 images of the *test* subset of the BSDS [16]. The images have a resolution of 321×481 pixels in grayscale. Each image is provided with 5 to 10 hand-made segmentations. Since those segmentations are given as region boundaries, we use them as ground truth in the quantification of the quality of the resulting edge images.

We have used in the experiments the MGED- T with $\Omega = \{0, 0.1, \dots, \sigma_M\}$, where σ_M is a parameter we have progressively increased. For the original GED- T we use the classical Gaussian smoothing with a single σ_M . In Fig. 5 we illustrate the evolution of TP and FP generated by the MGED- T and the GED- T using different values of σ_M and t-norm $T \in \{T_P, T_M\}$. We take as a FP any

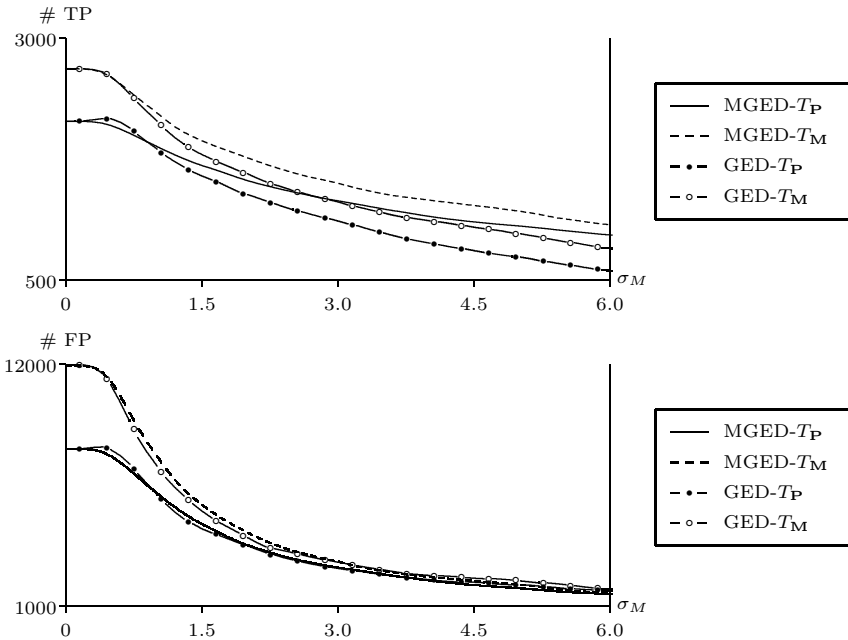


Fig. 5. Results obtained by the MGED- T and the GED- T on the sample images. The uppermost plot displays the average number of TP, while the second illustrates the average number of FP.

pixel being at most 2.5 positions away from a true pixel. Note that the σ_M stands for the standard deviation of G in the case of the GED- T and for the maximum σ when using MGED- T .

We observe in Fig. 5 that the number of FP is usually larger than the number of TP. Moreover, both quantities are reduced when σ_M increases. This is consistent with the examples in Figs. 2 and 4, where larger values of σ resulted in a lower number of edge points. The decrease of FP is due to the fact that noise and textures are usually high-frequency signals, and are therefore tackled by G_σ with relatively low σ . The decrease of TP is related to the fact that increasing smoothing may potentially displace the edges further away from its true position. Hence, the larger the value of σ , the more possibilities of the edges to be considered FP, even if related to a true edge. The number of TP can also decrease due to the overblurring of the image. Eventually, an edge may be smoothed to the point of not being detected (see the image in Fig. 3). The fact that an edge may become non-visible because of oversmoothing collides with the theoretical considerations by Konishi [10], but is very common in the practical application of multiscale methods.

We notice as well how the decrease of TP and FP cast different shapes. The decrease of FP is very fast at the beginning (when the high frequency noise is removed), and then decreases slowly. In the case of the TP, we observe a more

homogeneous behaviour, casting a slow decrease. In this way, even if the number of FP is much larger at the beginning (about 4 times higher in average), the largest values of σ produce a similar number of TP and FP. We also observe how the improvement of the MGED- T with respect to the GED- T increases along with σ . Obviously, when $n = 1$ we have that the result produced by each detector is the same. In the comparison of t-norms, we have that the T_M -based detectors always outperform their counterparts. Nevertheless, this is not the question raised in this experiment, since we intend to compare the original with the multiscale detectors.

Even if the improvement of the multiscale methods is evident, we have to bear in mind that it comes to the cost of a computational increase proportional to n . Therefore, we might find more interesting to use intermediate σ_M (in this case, for example, $\sigma_M = 3$) rather than using as many scales as possible.

4 Conclusions

We have introduced a multiscale extension of the GED- T , denoted MGED- T . In order to do so, we have used a fine-to-coarse edge tracking algorithm. Then we have illustrated the improvement it represents, with some visual examples. To conclude, we have tested the detector on a large number of images finding that, when increasing the amount of smoothing, the MGED- T provides a better preservation of the correctly detected edges while removing the spurious responses. However, it comes to the cost of a higher computational complexity, which might discard it in some scenarios.

References

1. Babaud, J., Witkin, A.P., Baudin, M., Duda, R.O.: Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 8(1), 26–33 (1986)
2. Baddeley, A.J.: Errors in binary images and an L^p version of the Hausdorff metric. *Nieuw Archief voor Wiskunde* 10, 157–183 (1992)
3. Canny, J.: A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 8(6), 679–698 (1986)
4. Carlotto, M.J.: Histogram analysis using a scale-space approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 9(1), 121–129 (1987)
5. Coleman, S., Scotney, B., Suganthan, S.: Multi-scale edge detection on range and intensity images. *Pattern Recognition* 44(4), 821–838 (2011)
6. Demigny, D.: On optimal linear filtering for edge detection. *IEEE Trans. on Image Processing* 11(7), 728–737 (2002)
7. Florack, L., Kuijper, A.: The topological structure of scale-space images. *Journal of Mathematical Imaging and Vision* 12, 65–79 (2000)
8. Heath, M., Sarkar, S., Sanocki, T., Bowyer, K.: A robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19(12), 1338–1359 (1997)

9. Jackway, P., Deriche, M.: Scale-space properties of the multiscale morphological dilation-erosion. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 18(1), 38–51 (1996)
10. Konishi, S., Yuille, A., Coughlan, J.: A statistical approach to multi-scale edge detection. *Image and Vision Computing* 21(1), 37–48 (2003)
11. Leung, Y., Zhang, J.S., Xu, Z.B.: Clustering by scale-space filtering. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22(12), 1396–1410 (2000)
12. Lindeberg, T.: Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision* 30(2), 117–156 (1998)
13. Lopez-Molina, C., Bustince, H., Fernandez, J., Couto, P., De Baets, B.: A gravitational approach to edge detection based on triangular norms. *Pattern Recognition* 43(11), 3730–3741 (2010)
14. Mallat, S., Hwang, W.: Singularity detection and processing with wavelets. *IEEE Trans. on Information Theory* 38(2), 617–643 (1992)
15. Marr, D., Hildreth, E.: Theory of edge detection. *Proceedings of the Royal Society of London* 207(1167), 187–217 (1980)
16. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings of the 8th International Conference on Computer Vision*, vol. 2, pp. 416–423 (2001)
17. McIlhagga, W.: The canny edge detector revisited. *International Journal of Computer Vision* 91, 251–261 (2011)
18. Peli, T., Malah, D.: A study of edge detection algorithms. *Computer Graphics and Image Processing* 20(1), 1–21 (1982)
19. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 12(7), 629–639 (1990)
20. Prewitt, J.M.S.: Object enhancement and extraction. In: *Picture Processing and Psychopictorics*, pp. 75–149. Academic Press (1970)
21. Qian, R., Huang, T.: Optimal edge detection in two-dimensional images. *IEEE Trans. on Image Processing* 5(7), 1215–1220 (1996)
22. Rosin, P.L.: Unimodal thresholding. *Pattern Recognition* 34(11), 2083–2096 (2001)
23. Russo, F.: FIRE operators for image processing. *Fuzzy Sets and Systems* 103(2), 265–275 (1999)
24. Shih, M.Y., Tseng, D.C.: A wavelet-based multiresolution edge detection and tracking. *Image and Vision Computing* 23(4), 441–451 (2005)
25. Sobel, I., Feldman, G.: A 3x3 isotropic gradient operator for image processing (1968); presented at a talk at the Stanford Artificial Intelligence Project
26. Sun, G., Liu, Q., Liu, Q., Ji, C., Li, X.: A novel approach for edge detection based on the theory of universal gravity. *Pattern Recognition* 40(10), 2766–2775 (2007)
27. Torre, V., Poggio, T.: On edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 8, 147–163 (1984)
28. Weickert, J.: *Anisotropic Diffusion in Image Processing*. ECMI Series, Teubner-Verlag (1998)
29. Witkin, A.P.: Scale-Space Filtering. In: *8th Int. Joint Conf. Artificial Intelligence*, Karlsruhe, vol. 2, pp. 1019–1022 (1983)
30. Yuille, A.L., Poggio, T.A.: Scaling theorems for zero crossings. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 8, 15–25 (1986)

A Study of the Suitability of Evolutionary Computation in 3D Modeling of Forensic Remains

José Santamaría^{1,*}, Oscar Cordon^{2,3}, Sergio Damas², José M. García-Torres⁴,
and Fernando Navarro⁵

¹ Dept. Computer Science, University of Jaén, Spain
jslopez@ujaen.es

² European Centre for Soft Computing, Asturias, Spain
{oscar.cordon,sergio.damas}@softcomputing.es

³ Dept. Computer Science and Artificial Intelligence, University of Granada, Spain
ocordon@decsai.ugr.es

⁴ SCI2S Research Group, University of Granada, Spain
jmgt@correo.ugr.es

⁵ Physical Anthropology Lab, University of Granada, Spain
fusely@ugr.es

Abstract. Image registration is a fundamental task in computer vision. Over the last decades, it has been applied to a broad range of situations from remote sensing to medical imaging, artificial vision, and CAD systems. In the last few years, there is an increasing interest in the application of the evolutionary computation paradigm to this task in order to solve the ever recurrent drawbacks of classical image registration methods. In this work, we will perform an experimental study on the performance of the most relevant evolutionary image registration methods proposed to date tackling a challenging real-world problem named 3D model reconstruction using laser range scanners. Specifically, we will make use of image datasets of human skulls provided by the Physical Anthropology Lab of the University of Granada, Spain.

Keywords: Image registration, evolutionary computation, 3D modeling.

1 Introduction

Image registration (IR) [20], is a crucial task in image processing systems. It is used to finding either a spatial *transformation* (e.g, rotation, translation, etc.) or a correspondence (matching of similar image entities) among two (or more) images taken under different conditions (at different times, using different sensors, from different viewpoints, or a combination of them), with the aim of overlaying

* This work is supported by the *Spanish Ministerio de Educación y Ciencia* (Ref. TIN2009-07727) including EDRF fundings and the *University of Jaén* (Ref. R1/12/2010/61) including fundings from *Caja Rural de Jaén*.

such images into a common one. Over the years, IR has been applied to a broad range of situations from remote sensing to medical imaging, artificial vision, and CAD systems. Different techniques have been independently studied resulting in a large body of research. In particular, the range image registration (RIR) problem is focused on the registration of images, named range images, acquired by laser range scanners [7].

IR is the process of finding the optimal spatial transformation (e.g, rigid, similarity, affine, etc.) achieving the best overlay between two (or more) different images. They both are related with the latter transformation, measured by a *Similarity metric* function. Such transformation estimation is interpreted into an iterative optimization procedure in order to properly explore the search space. Two search approaches have been considered in the IR literature: *matching-based*, where the optimization problem is intended to look for a set of correspondences of pairs of those more similar image entities in both the scene and the model images, from which the registration transformation is derived; and the *parameter-based*, where the strategy is to try to directly explore inside each range of the transformation parameters.

Aspects such as the presence of noise in images, image discretizations, orders of magnitude in the scale of the IR transformation parameters, the magnitude of the transformation to be estimated, etc., cause difficulties for traditional IR algorithms as the well-known *iterative closest point* (ICP) [2] algorithm, thus they become prone to get trapped in local minima.

In the last few years, the adoption of the *evolutionary computation* (EC) [1] paradigm has introduced an outstanding interest in the IR community in order to solve those problems due to their global optimization techniques nature. In particular, evolutionary algorithms (EAs) have been successfully applied for tackling the IR optimization process. The first attempts to solve IR using EC can be found in the early eighties [9]. Since then, several EC-based IR methods have been proposed to solve the IR problem.

In this work we introduce a practical study on the applicability of the EC paradigm for solving the IR problem. To do so, we consider some of the most relevant IR proposals making use of EC. Likewise, we will carry out an experimental study of the performance of these methods facing a real-world application of the IR problem named 3D object reconstruction using laser range scanners. In particular, we considered reconstructions of human skulls by using 3D images provided by the Physical Anthropology lab at the University of Granada (Spain).

The structure of this paper is as follows. First, Section 2 describes the IR problem and its specific application in the 3D model reconstruction of forensic objects using RIR methods. Next, Section 3 is devoted to introduce some of the most relevant IR methods using EC. Section 4 performs an experimental study by considering the previous introduced EC-based IR methods facing the real-world application of 3D model reconstruction of human skulls. Finally, Section 5 shows some conclusions of this work.

2 Preliminaries

2.1 Image Registration

There is not a universal design for a hypothetical IR method that could be applicable to all registration tasks, since various considerations on the particular application must be taken into account [20]. However, IR methods usually require the following four components (see Figure 1): two input **Images** named as Scene $I_s = \{p_1, p_2, \dots, p_n\}$ and Model $I_m = \{p'_1, p'_2, \dots, p'_m\}$, with p_i and p'_j being image points; a **Registration transformation** f , being a parametric function relating the two images; a **Similarity metric function** F , in order to measure a qualitative value of closeness or degree of fitting between the transformed scene image, noted $f'(I_s)$, and the model image; and an **Optimizer** that looks for the optimal transformation f inside the defined solution search space.

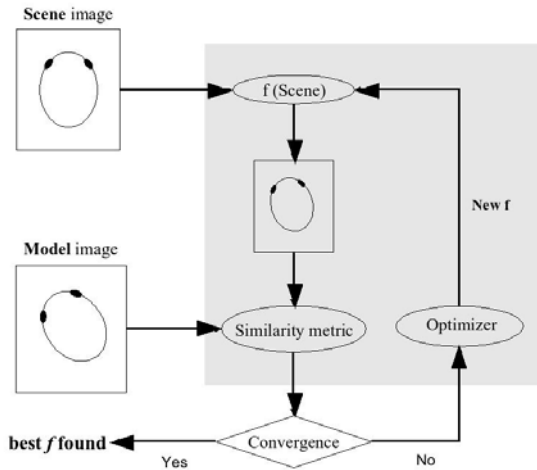


Fig. 1. The IR optimization process

Likewise, an iterative process is often followed until convergence, for instance, within a tolerance threshold of the concerned similarity metric. This is the case of the well-known ICP algorithm [2] which works as follows:

- A point set P is given with N_p points p_i from the scene image. The model image, X , is defined by N_x supporting geometric primitives: points, lines, or triangles.
- The iteration is initialized by setting $P_0 = P$, the registration transformation (using quaternions for rotations) by $q_0 = [1, 0, 0, 0, 0, 0]^t$, and $k = 0$. The next four steps are applied until convergence within a tolerance threshold $\tau > 0$:

1. Compute the matching (Y_k) between the current scene (P_k) and the model points (X) by the closest point assignment rule: $Y_k = C(P_k, X)$
2. Compute the registration transformation: $f_k(P_0, Y_k)$
3. Apply the registration transformation: $P_{k+1} = f_k(P_0)$
4. Terminate iteration when the change in mean square error (MSE) falls below τ

Despite the proposed scheme guarantees interesting properties such as fast and accurate convergence, it is strongly dependent on the initial estimation of the pose (transformation) between the images and it usually gets trapped in local optima. As we will demonstrate later, the application of EAs to the IR optimization process has caused an outstanding interest in the last few decades. Thanks to their global optimization nature, EAs aim to solve the latter drawback, not satisfactorily tackled by traditional IR methods.

2.2 3D Model Reconstruction Based on Range Image Registration

Range scanner devices are able to capture 3D images, named range images, from different viewpoints of the sensed object. Every range image partially recovers the complete geometry of the scanned object, then placing each of them in a different coordinate system. Thus, it is mandatory to consider a reconstruction technique to perform the accurate integration of the images in order to achieve a complete and reliable model of the physical object. This framework is usually called 3D model reconstruction and it is based on applying RIR techniques [17]. There are two RIR approaches to integrate multiple range images. The *accumulative* approach accomplishes successive applications of a pair-wise RIR method [1]. Once an accumulative RIR process is accomplished the *multiview* approach takes into account all the range images at the same time to perform a final global RIR step. Figure 2 depicts the steps of the 3D model reconstruction procedure when 3D models of human skulls are acquired.

As depicted in Figure 2, the 3D model reconstruction procedure carries out several pair-wise alignments of two adjacent range images in order to obtain the final 3D model of the physical object. Therefore, every pair-wise RIR method tries to find the Euclidean motion that brings the *scene* view (I_s) into the best possible alignment with the *model* view (I_m). We have considered an Euclidean motion based on a 3D rigid transformation (f) determined by seven real-coded parameters, that is: a rotation $R = (\theta, \textit{Axis}_x, \textit{Axis}_y, \textit{Axis}_z)$ and a translation $\mathbf{t} = (t_x, t_y, t_z)$, with θ and \mathbf{Axis} being the angle and axis of rotation, respectively. Then, the transformed points of the *Scene* view are denoted by

$$f(\mathbf{p}_i) = R(\mathbf{p}_i - \mathbf{C}_{I_s}) + \mathbf{C}_{I_s} + \mathbf{t}, \quad i = 1 \cdots N_{I_s} \quad (1)$$

where \mathbf{C}_{I_s} is the center of mass of I_s . We define the distance from a transformed I_s point $f(\mathbf{p}_i)$ to the *Model* view I_m as the squared Euclidean distance to the closest point \mathbf{q}_{cl} of I_m , $d_i^2 = \|f(\mathbf{p}_i) - \mathbf{q}_{cl}\|^2$.

¹ The use of the term *pair-wise* is commonly accepted to refer to the registration of pairs of adjacent range images.

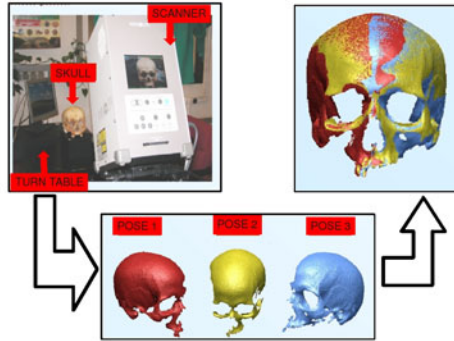


Fig. 2. 3D model reconstruction procedure

Hence, the RIR task can be formulated as an optimization problem developed to search for the Euclidean transformation f^* achieving the best overlapping of both images according to the considered *Similarity metric* F :

$$f^* = \underset{f}{\operatorname{arg\,min}} F(I_s, I_m; f) \quad \text{s.t.} : f^*(I_s) \cong I_m \quad (2)$$

Particularly, we used the median square error (MedSE) for tackling the RIR problem:

$$F(I_s, I_m; f) = \operatorname{MedSE}(d_i^2), \quad \forall i \in \{1, \dots, N_{I_s}\} \quad (3)$$

where $\operatorname{MedSE}()$ corresponds to the computation of the median d_i^2 value of the $N_{I_s}^{\text{th}}$ scene points. We have used the grid closest point (GCP) scheme ([19]) to speed up the computation of the closest point q_{cl} of I_m .

Finally, we have considered the feature-based RIR approach [20] in the subsequent experimental section. We used a 3D image processing algorithm in order to extract the most relevant features of the range images. These synthesized 3D images are used by the RIR method under study. We have followed the feature extraction procedure used in [17] to extract *crest lines* as salient features.

3 Evolutionary Image Registration

In the last few years, a new family of approximate algorithms is being extensively used by the IR community. They are named metaheuristics [10] and they are based on the extension of basic heuristics by considering their inclusion in an iterative process of improvement. One of the main advantage of these optimization alternatives is their capability to scape from local optima. That is one of the most relevant pitfalls of traditional IR methods (see Section 2.1).

As said, EC [1] is one of the most addressed approaches within metaheuristics. EC involves those strategies using computational models inspired on evolutive

procedures of nature as key elements in designing and developing of problem solving systems based on computers. In particular, the first attempts facing the IR problem using EC can be found in the eighties. Fitzpatrick et al. [9] proposed such approach using genetic algorithms (GAs) [11,13] to register 2D angiographic images in 1984. Since then, evolutionary IR has become a very active area and several well-known EAs have been considered to tackle the IR optimization process, causing an outstanding interest.

We have found the following evolutionary IR methods contributed in the last few years. Yamany *et al.* [19] used a GA based on the original binary representation of solutions proposed by Holland [11,13] facing the IR of 3D dental images; He and Narayana [12] tackled the IR of magnetic resonance images (MRIs) applying the explorative capabilities of the latter method by using more appropriated genetic operators together with a real-coded representation of solutions; Chow *et al.* [3] contributed with a new design of GA also using real-coded solutions and with the main novelty based on the inclusion of a restart mechanism named *dynamic boundary* in order to speed up the convergence of the algorithm tackling a RIR problem; Wachowiak et al. [18] contributed with a broad study on the performance of particle swarm optimization (PSO) [4,14] algorithms for solving the IR problem in biomedical applications, specifically registering single slices (2D images) of 3D volumes to whole 3D volumes of medical images; Cerdón *et al.*'s [6] proposal adapts the original binary scheme of the CHC [8] EA to a real-coded one and making use of characteristic information extracted from 3D MRIs; recently, Santamaría *et al.* [5] contributed with an enhanced extension of their previous proposal based on the scatter search (SS) [15] applied to RIR problems [17].

4 Computational Experiments

4.1 Experimental Design

The Physical Anthropology Lab of the University of Granada provided us three adjacent range images, I_1 , I_2 , and I_3 , of a human skull. The size (number of points) of every image is 76794, 68751, and 91590, respectively. Next, in order to follow the said feature-based RIR approach, we extracted crest lines features from each of these images, thus obtaining a reduced version of their original ones with 1181, 986, and 1322 number of points, respectively. Figure 3 shows the input 3D range images together with the result of applying the 3D crest line detector to every image

Finally, we configured two different RIR scenarios in order to accomplish the reconstruction of the 3D model: $RIR(I_1, I_2)$ and $RIR(I_3, I_2)$. Notice that the scene images I_1 and I_3 are aligned to the same model image, I_2 , that is considered as the anchor image.

4.2 Parameter Settings

All the methods presented in Section 3 have been run thirty different times. A different random rigid transformation is considered in every of the thirty

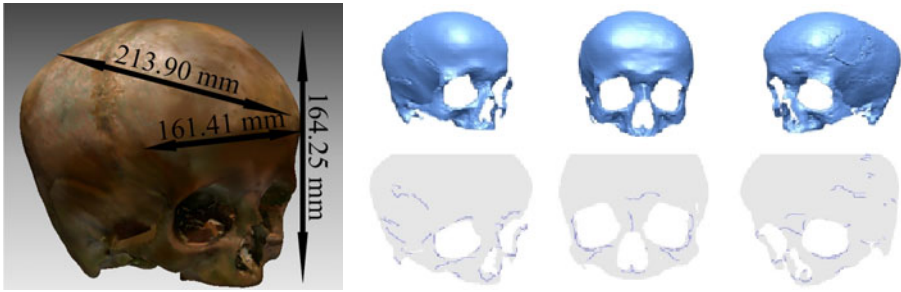


Fig. 3. From left to right: reconstructed 3D model and the three selected I_1 , I_2 , and I_3 range images acquired by a laser range scanner. Below the latter three is shown the resultant images after the application of the crest line detector.

runs. Thus, $(2 \times 30) = 60$ different RIR problem instances have been configured. We used a 2.6 GHz Intel Pentium IV CPU with 2GB RAM. We maintained the original parameter values of every IR method. On the other hand, we also used a recent enhanced version of the ICP algorithm [16] as a traditional (non metaheuristic-based) IR algorithm (see Section 2.1) for comparison purposes.

4.3 Analysis of Results

We used a turn table device (see Figure 2) with the aim to validate the reconstruction results estimated by the considered RIR methods. A ground-truth 3D model of the physical object is obtained using the latter mechanism. We considered the MSE metric in order to measure the quality of the RIR results:

$$MSE = \sum_{i=1}^r ||f(\mathbf{x}_i) - \mathbf{x}_i' ||^2 / r \tag{4}$$

where $f(\mathbf{x}_i)$ refers to the i^{th} transformed point of image scene using the estimated rigid transformation f , r is the image size of the latter one (before the application of the crest line detector), and \mathbf{x}_i' corresponds to the same i^{th} scene point in the ground-truth location.

Table 1. Statistics (from thirty different runs of every RIR method) of the considered RIR scenarios. In bold font are marked the best results according to minimum and mean values of MSE.

	RIR(I_1, I_2)			RIR(I_3, I_2)		
	Min.	Mean	Std. dev.	Min.	Mean	Std. dev.
Liu-ICP	159	9538	10185	2391	11334	8747
Yamany-GA	13	1884	4044	153	2691	4182
He-GA	9	93	73	75	872	1220
Chow-GA	43	1009	1013	117	2710	2130
Wachowiak-PSO	20	596	645	9	1608	4128
Cordón-CHC	18	248	780	131	1411	1495
Santamaría-SS	11	74	41	66	389	366

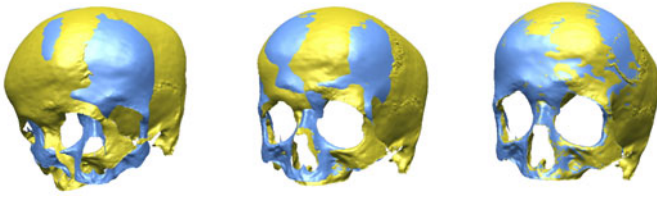


Fig. 4. RIR(I_3, I_2) scenario. From left to right: the first figure refers to the best estimation of Liu-ICP and the next two show the results provided by Santamaria-SS and its refined outcome by using Liu-ICP, respectively.

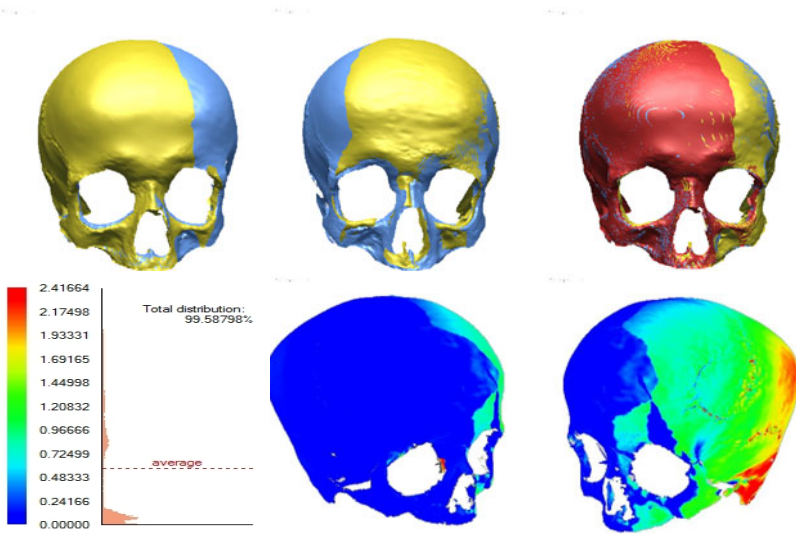


Fig. 5. From left to right: the top row shows the best two (pair-wise) prealignment IR results obtained by Wachowiak-PSO and the reconstruction result (combining the previous two prealignments) of the forensic dataset after refinement. The bottom row depicts the distance deviation histogram comparing the latter reconstruction result and the ground-truth 3D model (see Figure 3).

Table 1 presents the statistical results of the considered RIR scenarios. We remark the poor results achieved by the classical ICP-based algorithm, Liu-ICP, which is not able to address none of the considered RIR scenarios. This is due to the initial pose this algorithm should deal with, i.e. there is a high misalignment between both images. Moreover, all the evolutionary RIR methods outperform the results achieved by Liu-ICP according to both the best (minimum) and the mean MSE values. On the other hand, we highlight the low averaged performance (according to the mean MSE value) of the binary-coded GA (Yamany-GA) compared with the remaining evolutionary proposals which make use of more advanced schemes, e.g. using a real-coded representation of solutions. Among them, Santamaria-SS becomes the evolutionary RIR method

achieving the most accurate and robust outcomes due to the more suitable search strategy it makes use facing the RIR problem.

Some of the estimated 3D model reconstruction results are presented in Figure 4. On the other hand, the best outcome of Liu-ICP corresponds to a local optimum. On the other hand, Santamaria-SS is able to provide the refinement algorithm² (Liu-ICP) with an initial solution that converges to a near optimal RIR solution. Figure 5 shows these results in more detail.

5 Concluding Remarks

In the last few decades, the adoption of EC approaches have become a promising solution due to their behavior as global optimization techniques. They own a capability to perform robust search in complex and ill-defined problems as IR.

In the last few years, EC has been adopted in IR community to face some of the most challenging drawbacks of traditional methods. Evolutionary IR methods have demonstrated their good behavior facing the latter pitfalls. The main difficulty to be tackled is to find a reliable/robust manner to escape from locally optimal registration solutions. Several works reviewing the state of the art on IR/RIR methods have been contributed in the last years ([20]), but none of them addresses those IR contributions adopting an EA as optimization component. With the aim of bridging this gap, in this work we have introduced a preliminary study on, in our modest opinion, the most relevant state of the art evolutionary IR methods to date.

From the results obtained, we highlight the high performance and accurate results offered by the evolutionary RIR methods against those achieved by the traditional ones, when facing the 3D model reconstruction of human skulls. Nevertheless, the results presented in this contribution correspond to a preliminary study. Thus, we plan to extend this initial work considering a larger number of case studies together with including other state of the art evolutionary RIR methods.

Acknowledgments. We want to acknowledge all the team of the Physical Anthropology lab at the University of Granada (headed by Dr. Botella and Dr. Alemán) for their support during the acquisition of the specific range datasets.

References

1. Bäck, T., Fogel, D.B., Michalewicz, Z.: Handbook of Evolutionary Computation. IOP Publishing Ltd and Oxford University Press (1997)
2. Besl, P.J., McKay, N.D.: A method for registration of 3D shapes. IEEE T. Pattern Anal. Mach. Intell. 14, 239–256 (1992)

² Due to the evolutionary RIR approach usually obtains coarser results, a final refinement stage using ICP-based RIR algorithms is applied in order to obtain accurate outcomes.

3. Chow, C.K., Tsui, H.T., Lee, T.: Surface registration using a dynamic genetic algorithm. *Pattern Recogn.* 37, 105–117 (2004)
4. Clerc, M.: *Particle Swarm Optimization*. ISTE Publishing Company (2006)
5. Cerdón, O., Damas, S., Santamaría, J.: A Fast and Accurate Approach for 3D Image Registration using the Scatter Search Evolutionary Algorithm. *Pattern Recogn. Lett.* 27(11), 1191–1200 (2006)
6. Cerdón, O., Damas, S., Santamaría, J.: Feature-based image registration by means of the CHC evolutionary algorithm. *Image Vision Comput.* 22, 525–533 (2006)
7. Dalley, G., Flynn, P.: Range image registration: A software platform and empirical evaluation. In: *Third International Conference on 3-D Digital Imaging and Modeling (3DIM 2001)*, May 28– June 1, pp. 246–253 (2001)
8. Eshelman, L.J., Schaffer, J.D.: Preventing premature convergence by preventing incest. In: Belew, R., Booker, L.B. (eds.) *4th International Conference on Genetic Algorithms*, pp. 115–122. Morgan Kaufmann, San Mateo (1991)
9. Fitzpatrick, J., Grefenstette, J., Gucht, D.: Image registration by genetic search. In: *IEEE Southeast Conference*, Louisville, EEUU, pp. 460–464 (1984)
10. Glover, F., Kochenberger, G.A. (eds.): *Handbook of Metaheuristics*. Kluwer Academic Publishers (2003)
11. Goldberg, D.E.: *Genetic Algorithms in Search and Optimization*. Addison-Wesley, New York (1989)
12. He, R., Narayana, P.A.: Global optimization of mutual information: application to three-dimensional retrospective registration of magnetic resonance images. *Comput. Med. Imag. Grap.* 26, 277–292 (2002)
13. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor (1975)
14. Kennedy, J., Eberhart, R.: *Swarm Intelligence*. Morgan Kaufmann, San Francisco (2001)
15. Laguna, M., Martí, R.: *Scatter search: methodology and implementations in C*. Kluwer Academic Publishers, Boston (2003)
16. Liu, Y.: Improving ICP with easy implementation for free form surface matching. *Pattern Recogn.* 37(2), 211–226 (2004)
17. Santamaría, J., Cerdón, O., Damas, S., García-Torres, J., Quirin, A.: Performance evaluation of memetic approaches in 3D reconstruction of forensic objects. *Soft Comput.* 13(8-9), 883–904 (2009)
18. Wachowiak, M.P., Smolikova, R., Zheng, Y., Zurada, J.M., El-Maghraby, A.S.: An approach to multimodal biomedical image registration utilizing particle swarm optimization. *IEEE T. Evolut. Comput.* 8(3), 289–301 (2004)
19. Yamany, S.M., Ahmed, M.N., Farag, A.A.: A new genetic-based technique for matching 3D curves and surfaces. *Pattern Recogn.* 32, 1817–1820 (1999)
20. Zitová, B., Flusser, J.: Image registration methods: a survey. *Image Vision Comput.* 21, 977–1000 (2003)

L-System-Driven Self-assembly for Swarm Robotics*

Fidel Aznar, Mar Pujol, and Ramón Rizo

Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
{fidel,mar,rizo}@dccia.ua.es

Abstract. In this paper, an assembly swarm algorithm, that will generate microscopic rules from a macroscopic description of complex structures, will be presented. The global structure will be described in a formal way using L-systems (Lindenmayer systems). The proposed algorithm is mainly parallel and exhibit parsimony at microscopic level, being robust and adaptable. In addition, a comparison between a swarm with centralized control and our distributed swarm algorithm will be provided, comparing the time need by the swarm to be assembled and the number of messages exchanged between agents.

1 Introduction

Self-assembly, defined as the problem of designing local rules for a set of agents to be organized into a target structure (without pre-assign agent position or using centralized control) is a highly desirable feature in swarm robotics. As can be seen in several papers [2,8] self-assembly allows to develop tasks that a single agent cannot solve. Most self-assembly in nature happens in a non-centralized fashion. Like many natural processes (and unlike the operation of most software systems), distributed self-assembly can potentially exhibit properties such as parallelism (components act simultaneously, without being tied down by a centralized controller), parsimony (at the level of an individual component, behaviors are simple and elegant), robustness (the failure of a single controlling entity does not lead to the failure of the entire system) and adaptability (the system can respond to changing conditions) [4].

Although some methods have been developed to build structures assembling multiple agents, they usually do not allow the creation of complex ones, not exceeding 100 elements and generating fairly simple composed objects [2,8,1]. Other contributions use algorithms that are very difficult to be executed in a decentralized way and tend to generate the same number of rules as the assembled components. These rules were unable to parsimoniously capture any inherent order within a structure [5]. Moreover, some of these studies do not link global and local behaviours, which limits their application in a swarm-system.

* This work has been supported by the Ministerio de Ciencia e Innovacion, project TIN2009-10581.

In this paper, we will present a swarm algorithm that will be able to generate microscopic rules from a macroscopic description of complex structures, described in a formal way using L-systems. As we will see, the proposed algorithm is mainly parallel and exhibit parsimony at microscopic level being robust and adaptable. The communication needed by the swarm will be minimum compared with the size of the global structure to be generated. Moreover, a comparison between centralized control and the proposed distributed behaviour will be provided. Both, the time need by the swarm to be assembled and the number of messages exchanged between agents will be analyzed.

2 Self-assembly and Automatic Design

An innovative method of cooperation is achieved by self-assembly, that is, the capability of a group of mobile robots to autonomously connect to and disconnect from each other through some kind of device that allows physical connections. Self-assembly can enhance the efficiency of a group of autonomous cooperating robots in several different contexts. Generally speaking, self-assembly is advantageous anytime it allows a group of agents to cope with environmental conditions, which prevent them from carrying out their task individually. The design of the hardware and the control policies for self-assembling robots is a particularly challenging task. In the robotic literature, there are several types of hardware platforms composed of modules, which are capable of connecting to each other through some kind of connection mechanism. The majority of such systems fall into the category of self-reconfigurable robots [8]. However, as discussed before, most of these systems are not able to develop complex global structures usually not exceeding 100 elements and generating fairly simple composed objects, mainly due to the difficulty of linking microscopic and macroscopic behaviors.

There is a research field directly related to the creation of complex robotic structures that attempts to overcome the difficulties associated with designing and building robots. Automatic design, rather than try to create a general-purpose robot, propose that the morphology (and possibly the controller) will be obtained by applying evolutionary algorithms. Top used genetic representations in these algorithms are grammatical systems, L-systems, graphs and neural networks. Among all these alternatives, rewriting has proven to be a useful technique for defining complex objects by successively replacing parts of simple initial objects using a set of rewriting rules or productions. More specifically, L-systems are especially suitable for describing fractal structures, multicellular organisms or flowering stages of herbaceous plants and are used in theoretical biology for describing and simulating natural growth processes.

3 L-Systems and Turtle Interpretation

We previously stated that a L-system is a grammatical rewriting system which parallel applies rewriting rules to a string. Thus, more complex strings are

being generated from an initial axiom by successively repeating the character replacement process using a fixed set of transformation rules. We will focus on PD0L L-systems, that are mainly characterized by being independent of the context and were the first models used in biological developments. We could define a PD0L system as a triplet $G = (\Sigma, h, \omega)$, where Σ is an alphabet, h is an endomorphism defined on Σ^* , and ω is the axiom (an element of the Σ^*). The word sequence $E(G)$ generated by G consists of the words $h^0(\omega) = \omega, h(\omega), h^2(\omega), h^3(\omega), \dots, h^i(\omega)$, where i is a fixed number of iterations to be applied and h is nonerasing. The language of G is defined by $L(G) = \{h^i(\omega) | i \geq 0\}$

	$\omega : S$
	$h : S \rightarrow [A + B C]$
$\omega : F$	$A \rightarrow -D[A + A] - D - D$
$h : F \rightarrow FF + FF$	$B \rightarrow A + A$
	$D \rightarrow A + [D]$
	$C \rightarrow [C]$
a)	b)

Fig. 1. L-systems used in the experimental section. Alphabet Σ is not presented as can be observed in the production rules. The system b) uses a stack to *pop* and *push* the turtle status using the operators '[' and ']' respectively.

For example, given the following system $G = (\{a, +\}, \{h(a) = a + a\}, a)$, the generated language for G could be defined as $L(G) = \{(a + a)^{2^n} | n \geq 0\} = \{“a”, “a+a”, “a+a+a+a”, \dots\}$. We want to underline that h function is usually specified as production rules, so in the previous system h may be written as a rule such that $a \rightarrow a + a$. The language $L(G)$ is often used as input to a turtle interpreter which transform the language L into commands, in our case, this commans will be related to the movement and assembly of the robots.

We will define a turtle interpreted L-system as a four elements tuple $T = (\Sigma, h, \omega, m)$, where $m : \Sigma \rightarrow \mathcal{Y}$ is a function that maps symbols to commands, beeing \mathcal{Y} all the possible commands to be performed by the turtle interpreter. For the previous example m could be defined such that $m(a) = \text{createRobot}$ and $m(+)= \text{rotate}(\pi/2)$.

Such systems are used successfully for creating complex structures [6] due to its expression power and the existence of algorithms for its automatic generation [3]. However they are not directly applicable to swarm robotics, because the turtle language interpretation is sequential. In this way, the assembly would require a coordinated control which is incompatible with swarms that specifically tend to distributed control and individual agent parsimony.

In the next section, we will introduce a swarm algorithm capable of generating microscopic rules from a macroscopic structure specified by a system-L. The

process is performed in a distributed fashion using a subset of the robots of the swarm and minimizing the communication between agents. The result of this process will be the same structure specified by the sequential L-system obtained distributedly by the interaction of the robotic swarm.

4 L-System-Driven Self-assembly

When a PDL0 system T generates a language $L(T)$ it is interpreted by a turtle system in order to generate the global structure. This interpretation usually requires all language $L(T)$ to be generated. However, given the recursive nature and the context independence of PDL0 systems, we could try to avoid sequential symbol generation and try to build substructures for a given level i , and then assembly it to generate the final structure.

For all the experiments presented in this paper, we will use a swarm composed by a set of square robots able to assembly in any of its four faces. Any L-system used with this swarm will generate a composition of squares/robots in the space. Thus, function m could be defined using the following basic commands $m = \{\text{createRobot}, \text{turn}^+, \text{turn}^-\}$ where **createRobot** creates a robot at current turtle position and advance an unit, **turn+** and **turn-** turn the turtle $\pi/2$ and $-\pi/2$ radians respectively. Although these definitions are useful to understand the presented examples, it is important to underline that both, robot shape and the m function are independent from the algorithm presented below.

Giving a subset $\sigma = \{x|x \subseteq \Sigma \wedge m(x) = \text{createRobot}\}$, we can decompose the global structure in n substructures where n is the number of occurrences of σ in $L(T)$. Any symbol not in σ will be an operator (that will change turtle status) or a rule needed to specify the structure that will be ignored by the turtle.

Next, we will present the LSA (L-System Assembly) algorithm. LSA will require some method of communication between robots in order to specify and advertise the type, the state of the robots and to start the assembly process between two groups of agents. As the required information is limited, communication can be direct or indirect. The communication system is referenced in the algorithm using the RF object. It should be noted that each swarm robot is autonomous and must implement and run independently the LSA functionality specified in figure 1.

In this way, each robot will receive as input: a set of tokens to be processed (a tuple of symbols from Σ) which contain at most one nonterminal symbol (without loss of generality, we assume that $|\omega| = 1$), a set of ancestors nodes and the level i where a robot is currently working. If a symbol is terminal or has reached the maximum number of iterations, then the robot will execute the turtle task specified by m (line 4) and will look for a robot to assemble (or if it has completely assembled the current level it will continue the algorithm at level $i - 1$ with its next ancestor node). Otherwise the robot will expand all non-terminal symbols using h and request the participation of new robots (line 12) to process it. Next, the robot is released from its task (line 15) and therefore, is free to perform any other task in the swarm. In figure 2 a more detailed trace of the algorithm for the L-system depicted in figure 1a is presented.

Algorithm 1. LSA: L-System assembly

```

input :  $tg \in TkGroup$ ,  $ancestor \in List[Node]$ ,  $level \in \mathbb{Z}$ 
1 begin
2   val RF  $\in RobotFactory$ 
3   if  $level = systemL.numIt$  or  $\neg hasRule(tg)$  then
4     | Process( $ancestors.head$ )
5   else
6     | val rTK  $\leftarrow tokenizeRule(expandRule(getRule(tg)))$ 
7     | foreach  $t$  of  $TkGroup \in rTK$  do
8       | val cState  $\leftarrow$  if  $|rTK| > 0$  then {if  $t.isLeft$  then :active: else
9         | :passive:} else :up:
10      | val  $n \leftarrow$  new Node( $t, t.kind, rTK, pos(t, rTK)+1, ancestors, cState,$ 
11        |  $level+1$ )
12      | RF.AskNewRobot.LSA( $t, push(n, ancestors), level+1$ )
13    | end
14  | end
15  | RF.FreeRobot(this);
16 end

```

Moreover, when a robot will start an assembly process it must be able to run a set of local turtle command to create progressively the global assembled structure. Each assembled agent save two vectors, one to specify the initial position of the assembly and another to indicate the final turtle position. In this way, agents can develop local assembly tasks and then continue at higher levels with no coordination problems. A detailed example of a turtle interpreter for the previous L-system is analyzed in figure 3.

5 Experimentation

In this section, we will comment the generation of various global structures using sequential generation (default L-systems) and the distributed LSA algorithm. More specifically, we will compare the time required for assembly completion and the number of messages required for agent communication. We will assume an initial random robot arrangement, so we will not consider the displacement of robots in the assembly time. In this way, we define t_e as the average time needed to establish an assembly between two groups of robots.

Two PDL0 systems, presented in figure 1, will analyzed. System 1a has only a rule F and a terminal symbol $+$, where $\mathcal{Y} = \{F\}$. System 1b consist of five rules, where $\mathcal{Y} = \{A, B, C, D\}$ and $+$, $-$ are terminal symbols. This is a bracketed PDL0 system that uses a stack to store turtle positions. Our algorithm is able to work with this kind of systems with no modification. More information about bracket L-systems can be found here 3. Figure 4 shows a comparison of the assembly time required to generate the overall structure (measured in units of

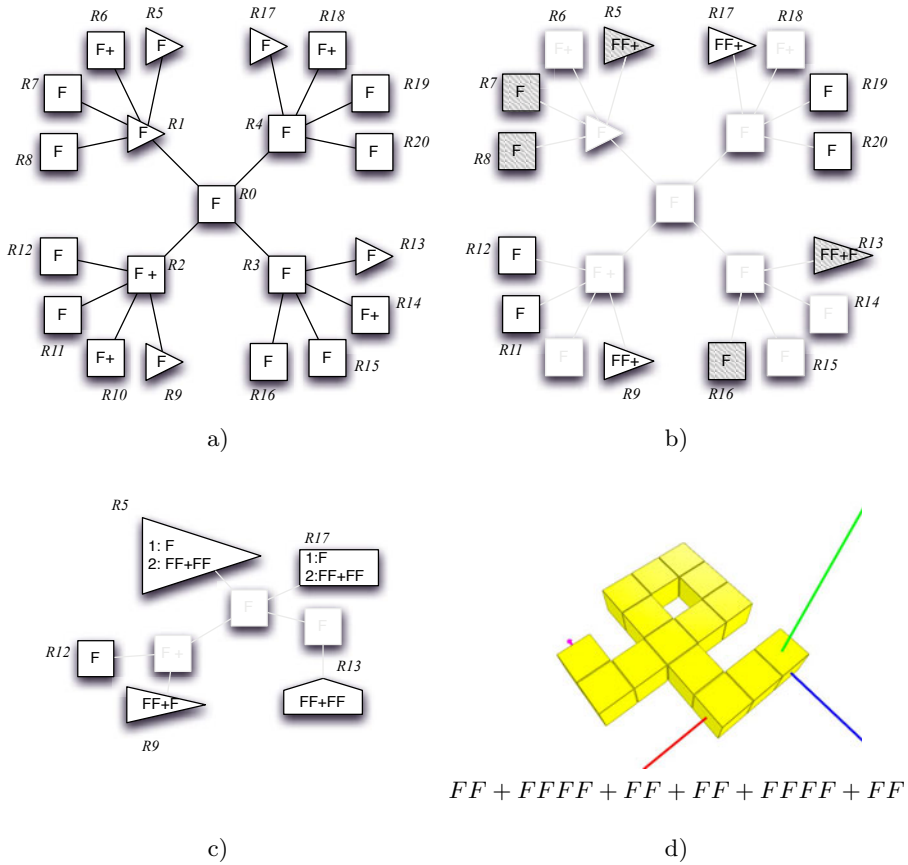


Fig. 2. Trace of the LSA algorithm for the L-system presented at figure 11a for $i = 2$. In a) the order of recruiting robots is shown: first, R_0 decides to start the assembly process and therefore, starts processing the axiom $\Omega = F$, expanding it using h (line 6 of the algorithm). Four expanded symbols belong to Υ so that R_0 will recruit 4 more swarm robots. Finished the recruiting task, R_0 is released being fully available for the swarm (line 15). The same process is developed by robots $R_1 \dots 4$. The descendants of previous nodes are considered terminals (since they have reached the max iteration level) so that they must fully develop the assembly task (line 4). The leftmost node of a descendant is considered to be active, other nodes will be passive. In b) the robots begin the assembly process. Active robot R_5 obtain the form/kind of its future match (using h and his current token) ' $F +$ ' and looks for it. Any passive robot with this form and with the same ancestors of R_5 could be a good candidate (in this case both, R_{14} and R_6 could be assembly partners of R_5). Once selected its partner and assembled, R_5 updates its form to ' $FF +$ '. In parallel, the controller of its assembly partner is released (line 10 of **process** function). This process will be repeated for all robots at the same level until no more assembly is required. We could observe at c) that R_{13} has no brothers so that its node is marked as upper and must level up to continue the assembly process for level-1 (line 3 of **Process** function). In this current level, R_5 will look for a partner assembling with any robot of the form ' $F +$ '. The result of this process is shown in d), where each cube represents a swarm robot.

Function. $Process(n \in Node)$

```

1 while  $\neg RF.isReleased(this)$  and  $\neg finish$  do
2   switch  $n.state$  do
3     case  $:up$ :  $n.state \leftarrow Upperize(n)$ 
4     case  $:passive$ :  $RF.PublishRobotKind(this)$ 
5     case  $:active$ :
6       while  $n.state.isActive$  or  $Error$  do
7         val  $mate = RF.findMate(n)$ 
8         if  $mate$  then
9            $[n.state, finish] \leftarrow RF.fuse(n, mate)$ 
10           $RF.Release(mate)$ 
11        end
12      end
13      if  $Error$  then  $ProcessError()$ 
14    endsw
15  endsw
16 end

```

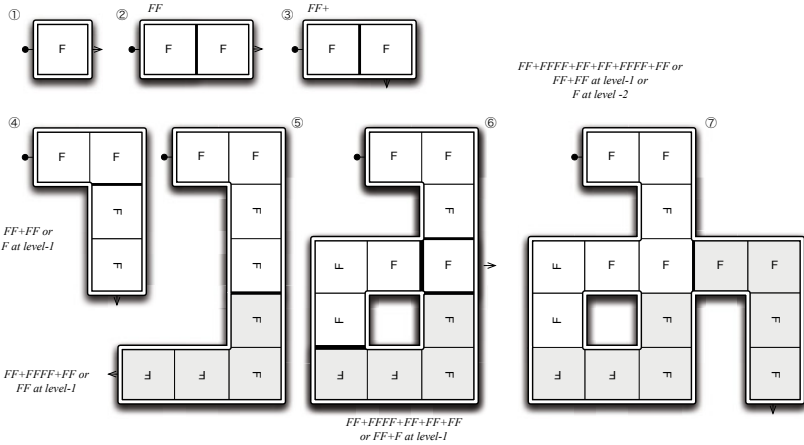


Fig. 3. Turtle system trace for the L-system presented in figure 1. 1) A terminal node robot saves its form “F” and sets its initial and final turtle position vector (circle and arrow respectively). 2) A robot “F” assembles with another robot of a kind “F+”. This process is as easy as to join the arrow of the active node with the circle of the passive node “FF”. 3) Then, turtle vector is updated by the symbol ‘+’, rotating $\pi/2$ the turtle angle “FF+”. 4) An assembled robot “FF+” is joined with another of the form “FF”. In 5) we see the same process for two robots of a kind “FF+FF” (at level 2). It is important to remark that assembling level 2 “FF+FF” robots is the same as assembling level 1 “F” robots (“F” at level 1 is expanded to “FF+FF” at level 2). In 6) we could observe the result of assembling a level 1 robot with form “FF+” with other robot of a kind “F”. In 7) the final assembly structure is presented.

t_e) and the number of messages exchanged between swarm agents for a given number of iterations.

The first important result we found for the two PDL0 analyzed, related to assembly time, is that a swarm running the LSA algorithm is an order of magnitude more efficient than sequential assembly. This is an expected result because for each level, LSA swarm acts recruiting and assembling robots in a parallel fashion, in contrast to the sequential generation. However, the recursion i substantially affect the assembly process and thus, for systems with small i or little recursion rules, the assembly time would not vary so much.

Moreover, the number of messages exchanged are bigger for the LSA swarm than for the sequential one. We would expect twice message traffic from the distributed approach as from centralized one, mainly because each robot has to communicate its type and has to negotiate with other robots for assembly. However, we believe that the number of messages is quite reduced so that the data could be sent even using indirect communication, such as color states (see [8]). For example, the number of robots that define the structure of $\mathbb{I}b$ for $i = 7$ is more than 16000, and about 81810 messages are generated by the LSA swarm (a mean of 5 messages is used for each robot to communicate). On the other hand, we need a minimum of two messages per robot to coordinate the assembling process for the sequential approach.

6 Discussion

In this paper, a swarm algorithm that is able to generate microscopic swarm interactions to build a complex global structure, previously specified using PD0L systems has been presented. The result is a robotic swarm without centralized control, able to build a given structure where individual agents only use local information showing individual parsimony. The global structure may be specified by the programmer or may be even learned by the swarm [3]. We have shown that a swarm following the LSA algorithm is able to assemble an order of magnitude faster than using a centralized control. It has also been tested that the required information to be transmitted for each of the individual agents is minimal compared to the size of the structure to be generated.

We would like to conclude emphasizing some points about the presented approach. On the one hand, for a given system PD0L G we can find an equivalent G' that generates the same language $L(G) = L(G')$. However, this equivalence does not mean that the proposed LSA algorithm develops the assembly process in the same way. In fact, this deals with a very interesting problem: how to find equivalent PD0L systems that guide the assembly process optimally.

On the other hand, it has been assumed that the number of robots that will be requested to complete the assembly process will be fixed and will be exactly the necessary to generate the final structure. To be dependent on a small set of robots or not to be scalable with the size of the swarm is not desired in swarm robotics. It is possible to modify the proposed algorithm to take into account any robot failure. In line 13 of `process` function it is easy to request a

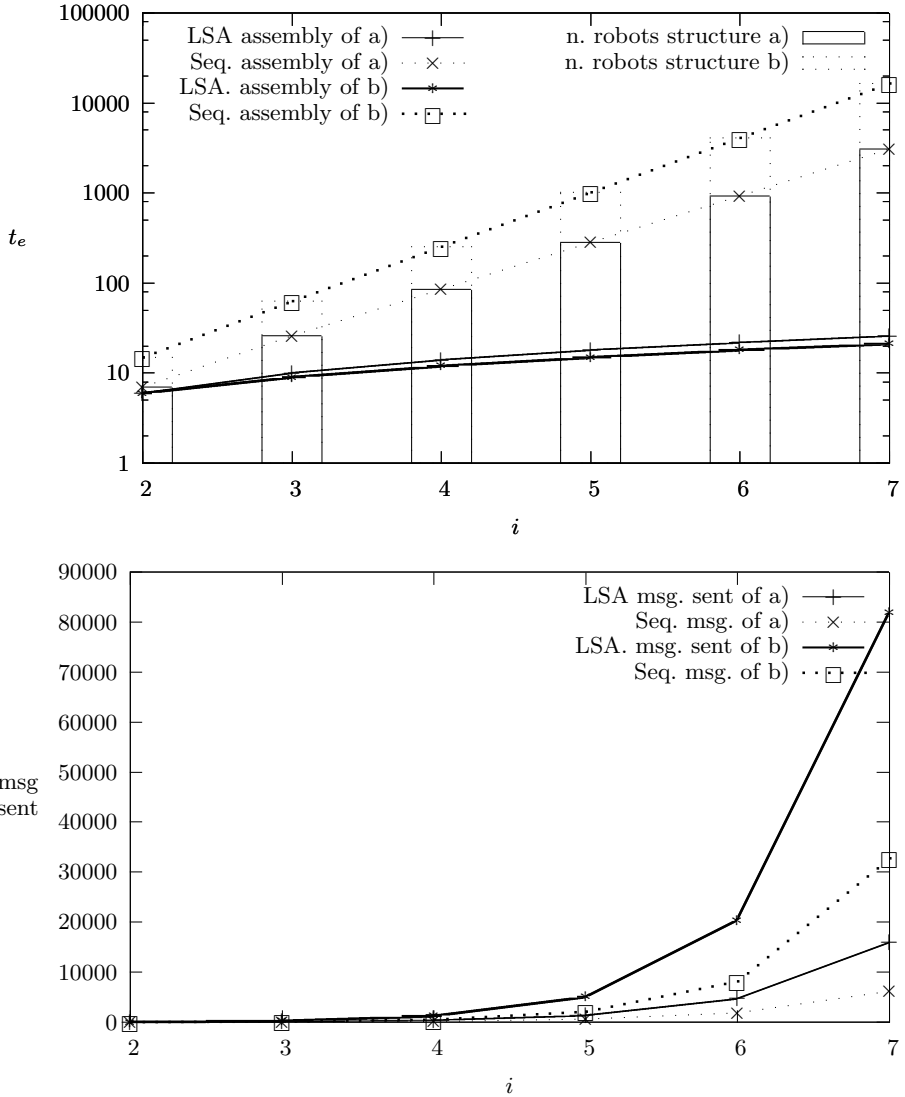


Fig. 4. Comparison of assembly time (top) and sent messages (below) for LSA and sequential swarm system. The number of robots that define the final structure for each iteration level i is also shown. We want to note that axis t_e uses a logarithmic scale.

robot replacement if the agent could not find its partner in order to successfully complete the assembly process. Another interesting point, is the generation of PDL0 to be more insensitive to individual components failure. It is important to remember that the proposed algorithm does not place any agent in fixed positions, but rather an agent of a particular form may be matched by any other robot that looks for this form. In this way, a robotic failure will not be visible until most of the structure has been constructed. This leads that individual failures or even an insufficient number of robots on well designed PDL0 only affects the final structure minimally making it slightly different but yet useful for the swarm.

We want to underline that exist several studies that establish how to obtain a L-system to generate a given structure [7]. This provides to the swarm developer a simple mechanism to design the necessary types of assembly.

Finally, we want to point a major problem that has not been addressed in this article but that we consider to be the next step to be taken into account: how to coordinate individual robotic movements as part of an assembled structure? Although there are studies that address these issues [8], [2], it is a relatively new research field, where most of the analyzed structures are very simple and do not require complex management.

References

1. Ampatzis, C., Tuci, E., Trianni, V., Christensen, A.L., Dorigo, M.: Evolving self-assembly in autonomous homogeneous robots: Experiments with two physical robots. *Artif. Life* 15, 465–484 (2009)
2. Baldassarre, G., Parisi, D., Nolfi, S.: *Distributed Coordination of Simulated Robots Based on Self-Organization*, vol. 12. MIT Press, Cambridge (2006)
3. Farooq, H., Zakaria, M.N., Hassan, M. F., Sulaiman, S.: An Approach to Derive Parametric L-system Using Genetic Algorithm. In: Badioze Zaman, H., Robinson, P., Petrou, M., Olivier, P., Schröder, H., Shih, T.K. (eds.) *IVIC 2009*. LNCS, vol. 5857, pp. 455–466. Springer, Heidelberg (2009)
4. Groß, R., Dorigo, M.: Self-assembly at the macroscopic scale. *Proceedings of the IEEE* 96(9), 1490–1508 (2008)
5. Grushin, A., Reggia, J.A.: Parsimonious rule generation for a nature-inspired approach to self-assembly. *ACM Trans. Auton. Adapt. Syst.* 5, 12:1–12:24 (2010)
6. Kurth, W.: Specification of morphological models with l-systems and relational growth grammars. *Image, Journal of Interdisciplinary Image Science* 5 Special Issue (2007)
7. St’ava, O., Benes, B., Mech, R., Aliaga, D.G., Kristof, P.: Inverse procedural modeling by automatic generation of l-systems. *Computer Graphics Forum* 29(2), 665–674 (2010)
8. Tuci, E., Groß, R., Trianni, V., Mondada, F., Bonani, M., Dorigo, M.: Cooperation through self-assembly in multi-robot systems. *ACM Trans. Auton. Adapt. Syst.* 1, 115–150 (2006)

An Study on Ear Detection and Its Applications to Face Detection*

Modesto Castrillón-Santana, Javier Lorenzo-Navarro,
and Daniel Hernández-Sosa

SIANI
Campus de Tafira
Universidad de Las Palmas de Gran Canaria
35017 - Spain

Abstract. OpenCV includes different object detectors based on the Viola-Jones framework. Most of them are specialized to deal with the frontal face pattern and its inner elements: eyes, nose, and mouth. In this paper, we focus on the ear pattern detection, particularly when a head profile or almost profile view is present in the image. We aim at creating real-time ear detectors based on the general object detection framework provided with OpenCV. After training classifiers to detect left ears, right ears, and ears in general, the performance achieved is valid to be used to feed not only a head pose estimation system but also other applications such as those based on ear biometrics.

Keywords: face detection, facial feature detection, ear detection, Viola-Jones.

1 Introduction

Among the wide literature on the face detection problem, the well known Viola-Jones face detector [21] has received lots of attention. This interest is justified not only thanks to its remarkable performance, but also due to its availability to a large community via the OpenCV library [13,14].

However, Viola and Jones [21] designed a general object detection framework. The approach is therefore suitable to be applied not only to the face pattern. Indeed, several researchers have already trained classifiers to detect different targets, and distributed to the community in the current OpenCV release [7,18]. Among those available classification cascades, it is observed that most of them are focused on the frontal face and its inner facial features (eyes, mouth and nose). There are two exceptions within the available classifiers in OpenCV, but also related with human detection, these are the profile face detector [2], and the head and shoulders [12] detector. Both are particularly less reliable than those designed for the frontal pose [3].

* Work partially funded by the Spanish Ministry of Science and Innovation funds (TIN 2008-06068).

In this paper, we are interested in putting in practice the Viola-Jones framework to automatically detect the ear pattern in images. It is evident that the ear pattern, as present in Figure 1, would be visible only if it is not occluded and the head is not frontal, i.e. the head presents a profile (or almost) pose.

Automatic ear detection is indeed an useful ability in the human machine interaction scenario. Its detection can for instance be applied in conjunction with a face detector to better fit a 3D head model onto an individual, achieving better pose estimation [16,20]. Additionally, the ear pattern has been used in biometrics by its own or as supplementary cue [10]. Therefore, its live and automatic detection would be of interest to multimodal recognition based on ear and face images [4,9].

To reach this objective we have adopted, as mentioned above, the Viola-Jones framework. Indeed the AdaBoost approach has already been used to design ear detectors [1,8]. The main difference with both works is that we are employing standard tools integrated with OpenCV to create the cascade. Our final aim is to make the classifiers available, including them in OpenCV, to serve as baseline. We consider that this is an advantage as previous researchers have provided their results but not released their detectors to the community. Abaza et al. are also concerned about reducing the training time. We will see later that the processing time is not so high using the standard OpenCV commands. Additionally, we present a larger experimental setup, in comparison to [8] and less restricted, in terms of controlled imagery, if compared to [1].

Section 2 summarizes the Viola-Jones detection framework. Section 3 describes the data used for the experimental setup. Results and conclusions are presented in sections 4 and 5, respectively.

2 Viola-Jones General Object Detection Framework

Creating a detector with the Viola-Jones framework requires: 1) a large training set (at least some thousands) of roughly aligned images of the object to detect or target (positive samples), and 2) another even larger set of images not containing the target (negative samples). This setup is a tedious, slow and costly phase, that has been summarized in different brief tutorials, e.g. [19].

Both images sets are used to train a boosted cascade of weak and simple classifiers. The main idea behind this framework is to apply less effort in processing the image. Each weak classifier is fast and has the ability to provide a high detection ratio, with a small true reject ratio, i.e. it is able to detect the target most of the time. However, a weak classifier is not able to reject all the patterns without interest. Indeed, it would be enough if it is able to reject half of them. In terms of execution time, the resulting cascade of classifiers would be much faster than a strong classifier with similar detection rates.

Each weak classifier uses a set of Haar-like features, acting as a filter chain. Only those image regions that manage to pass through all the stages of the detector are considered as containing the target. For each stage in the cascade, see Figure 2, a separate subclassifier is trained to detect almost all target objects

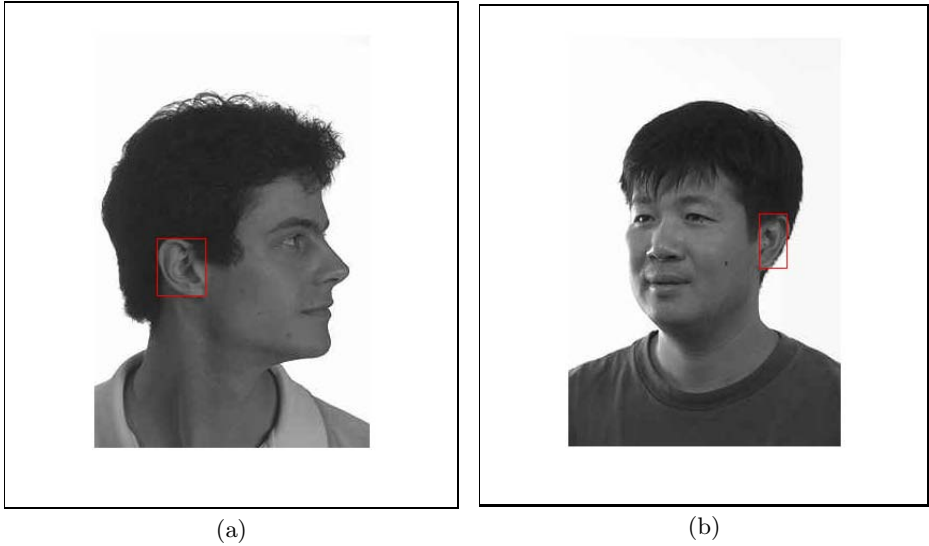


Fig. 1. FERET dataset [17] ear annotation examples. If mostly visible, the ear pattern has been annotated manually rotated in faces rotated along the vertical axis (out-of-plane rotation.)

while rejecting a certain fraction of those non-object patterns that have been incorrectly accepted by previous stage classifiers.

Theoretically for a cascade of K independent weak classifiers, the resulting detection rate, D , and the false positive rate, F , of the cascade are given by the combination of each single stage classifier rates:

$$D = \prod_{i=1}^K d_i \quad F = \prod_{i=1}^K f_i \quad (1)$$

Each stage classifier is selected considering a combination of features which are computed on the integral image, see Figure 3a-b. These features are reminiscent of Haar wavelets and early features of the human visual pathway such as center-surround and directional responses, see Figure 3c. The implementation [15] integrated in OpenCV [7] extended the original feature set [21].

With this approach, given a 20 stage detector designed for refusing at each stage 50% of the non-object patterns (target false positive rate) while falsely eliminating only 0.1% of the object patterns (target detection rate), its expected overall detection rate is $0.999^{20} \approx 0.98$ with a false positive rate of $0.5^{20} \approx 0.9 \times 10^{-6}$. This schema allows a high image processing rate, due to the fact that background regions of the image are quickly discarded, while spending more time on promising object-like regions. Thus, the detector designer chooses the desired number of stages, the target false positive rate and the target detection rate per stage, achieving a trade-off between accuracy and speed for the resulting classifier.

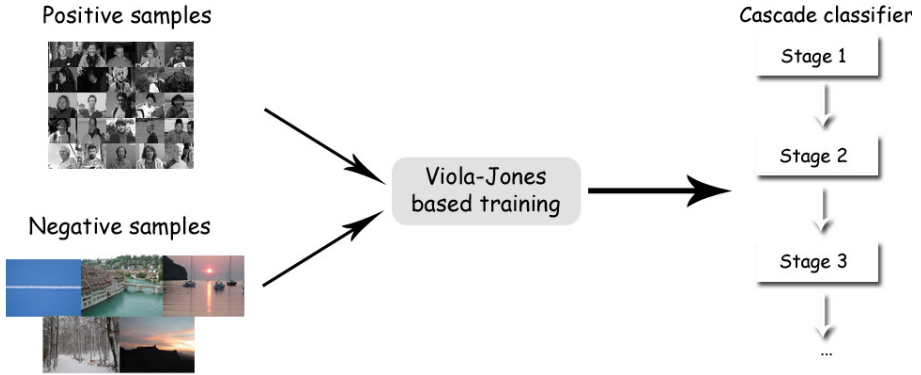


Fig. 2. Typical training procedure for a Viola-Jones’ based classifier. Each classifier stage is obtained using positive and negative samples accepted by the previous stage. Adapted from [3].

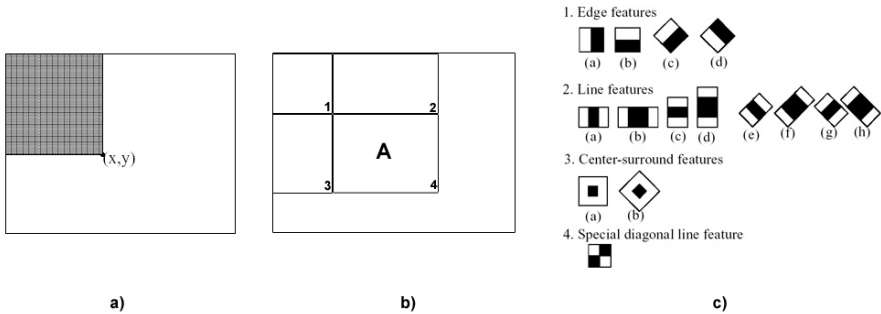


Fig. 3. a) The Integral Image stores integrals over subregions of the image. b) The sum of pixel values in A is $(x_4, y_4) - (x_2, y_2) - (x_3, y_3) + (x_1, y_1)$ [5]. c) Features prototypes considered in the implementation integrated in OpenCV [13,15].

Given an input image, the resulting classifier will report the presence and location, in terms of rectangular container, of the object(s) of interest in the image.

The availability of different tutorials, e.g. [19], guides OpenCV users to collect, annotate and structure the data before building the different classifier training. To test their performance, they must later be tested with an independent set of images.

3 Datasets

The imagery used to train and evaluate the ear detection performance is the FERET dataset [17]. Even when this dataset is mainly known in the face

recognition literature, it is used in this paper to evaluate the ear detection performance. The dataset contains two subsets that we refer below as FERETCD1 and FERETCD2.

The dataset includes frontal, profile and inbetween faces. For our purpose, we have considered just the profile or almost profile images contained in the thumbnails folder of both subsets. Each ear present in those images, has been manually annotated defining a container with four points as seen in Figure 1.

As the reader can observe in Figure 1, some annotated ears will correspond to the left and some to the right ear. We have created three different classifiers to detect ears (left ear, right ear, just ear). For that purpose, we have flipped all the annotated images in FERETCD1 to build a larger training set suitable to train the different patterns. Those annotated images contained in FERETCD1 constitute the set of positive samples. The number of annotated images in both sets is reflected in Table 1. The set of negative samples (also been flipped to avoid any bias) is composed mainly of large wallpaper images that do not contain the target pattern. These datasets are used to train the different ear detectors.

The FERETCD2 subset is used for evaluation. The resulting classifiers provide detection results, that must be compared with the annotation data to determine the classifier goodness. The criterion adopted to consider an ear detection, e_d , as true detection, will observe the overlap with the annotated container, e_a , and the distance between both containers:

$$\text{correct detection} = \text{overlap OR close} \quad (2)$$

where *overlap* is

$$\text{overlap} = \begin{cases} \text{true} & \text{if } \frac{a_a \cap a_d}{a_a} > 0.5 \\ \text{false} & \text{otherwise} \end{cases} \quad (3)$$

being a_a and a_d the area of the annotated and detected container. And *close* is defined as

$$\text{close} = \begin{cases} \text{true} & \text{if } \text{dist}(e_d, e_a) < 0.25 \times e_a\text{-width} \\ \text{false} & \text{otherwise} \end{cases} \quad (4)$$

where *dist* refers to the distance between both container centers.

Table 1. Total number of images contained in each subset, and the number of ears annotated in each set (observe that not all the images present a profile or almost profile pose). Hidden ears have not been annotated, but some partially hidden ears have been approximately estimated.

Set	Total number of images	Annotated ears
FERETCD1	5033	2798
FERETCD2	4394	1947

4 Experimental Results

4.1 Ear Detection Performance

As mentioned above, we have used the FERETCD1 and the negative images sets to train the different target classifiers, while the FERETCD2 set has been used to evaluate both classifiers.

Giving some training details, on one side, the number of positive samples used to create each classifier based on the OpenCV implementation was 3000 (6000 for the ear detector). The reader can observe that this number is slightly larger than the number of positive samples indicated in Table I. Indeed, the utility integrated in OpenCV to compile the file of positive samples creates additional training samples making use of reduced affine transformations. On the other side, 10000 was the number fixed as negative samples. The rest of the training parameters employed were mainly default values, excepting the pattern size selected, 12×20 , and the tag indicating that the target pattern is not symmetric.

The training time to compute each 20 stages classifiers, using a 2.66Ghz processor, was around 30 hours for the left and right ear detectors, and 40 hours for the general ear detector.

The detection results achieved for the FERETCD2 set are presented in Figure 4a. For each classifier, its receiver operating characteristic (ROC) curve was computed applying first the 20 stages of each classifier, and four variants reducing its number of stages (18, 16, 14 and 12 respectively). Theoretically, this action must increase both correct, D , and false, F , detection rates. The precise positive and negative detection rates for both specialized classifiers using 20, 18, 16 and 14 stages are presented in Table II.

Observing the figure, it is evident that the specialized detectors, i.e. those trained to detect only the left or only the right ear, perform better. For similar

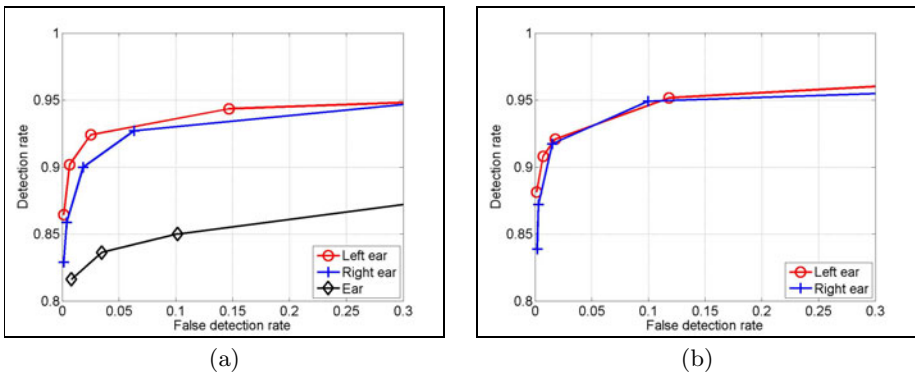


Fig. 4. (a) Left and right ear detection results, training with CD1 and testing with CD2. (b) Left and right ear detection results, training with CD1 and CD2, and testing with CD2.

detection error rates, e.g. 5% the detection is around 92% and only 84% for the ear detector. The precise results presented in the table for the left and right ear detectors, suggest that both detectors do not perform exactly the same. Indeed, the left ear detector seems to have a lower false detection rate for similar positive detection rate. This effect can be justified by the fact that the false negative samples selection integrates some random decision during the training phase.

In summary, both specialized classifiers perform remarkably well for this scenario, while keeping a low false detection rate. In fact both detectors locate correctly more than 92% of the target patterns presenting an error rate around 5%. They are therefore robust ear detectors in the experimental setup. To process the 1947 images contained in the FERETCD2 set, even when their size is not homogeneous, the average processing time was 45 and 48 milliseconds respectively for the right and left detector. Figure 4b, presents the results achieved training with both subsets and testing with the FERETCD2 set.

In addition, we have tested the detectors with real video using a 640×480 webcam achieving close to real-time performance. This is achieved even when no temporal information is used to speed up the processing. The detectors are therefore valid to be applied for interactive purposes.

Table 2. Detection results using 20, 18, 16 and 14 stages

Approach	Detection rate	False detection rate
Left ear (20)	0.8644	0.0015
Left ear (18)	0.9019	0.0067
Left ear (16)	0.9240	0.0252
Left ear (16)	0.9435	0.1469
Right ear (20)	0.8290	0.0015
Right ear (18)	0.8588	0.0041
Right ear (16)	0.8998	0.0185
Right ear (16)	0.9271	0.0632

4.2 Face Detection Improvement

To illustrate the interest of the facial features detection ability in conjunction with a face detector, we have performed a brief analysis on the the FDDB (Face Detection Data Set and Benchmark) dataset [11]. This dataset has been designed to study the problem of real face detection. The dataset contains a 5171 annotated faces taken from the Faces in the Wild dataset [6].

On that dataset we have applied face detection using two different approaches:

- Face detection using an available in OpenCV detector.
- Face detection using an available in OpenCV detector, but confirming the presence of at least 2 inner facial features (eyes, nose, mouth, and ears) using the facial feature detectors present in OpenCV, plus our ear detectors.

The additional restriction imposed forces the location for a face candidate (detected by a face detector) of at least two inner facial features. The main benefit

is that the risk of false detections is reduced as reflected in Table 3. However, the main benefit of the ear detection inclusion, is that when an ear is detected it additionally provides an evidence about the head pose, this is illustrated in Figure 5.

Table 3. Face detection results on the Fddb set

Approach	Detection rate	False detection rate
Face detection	71.55	6.57
Face detection and 6 FFs	65.94	1.85

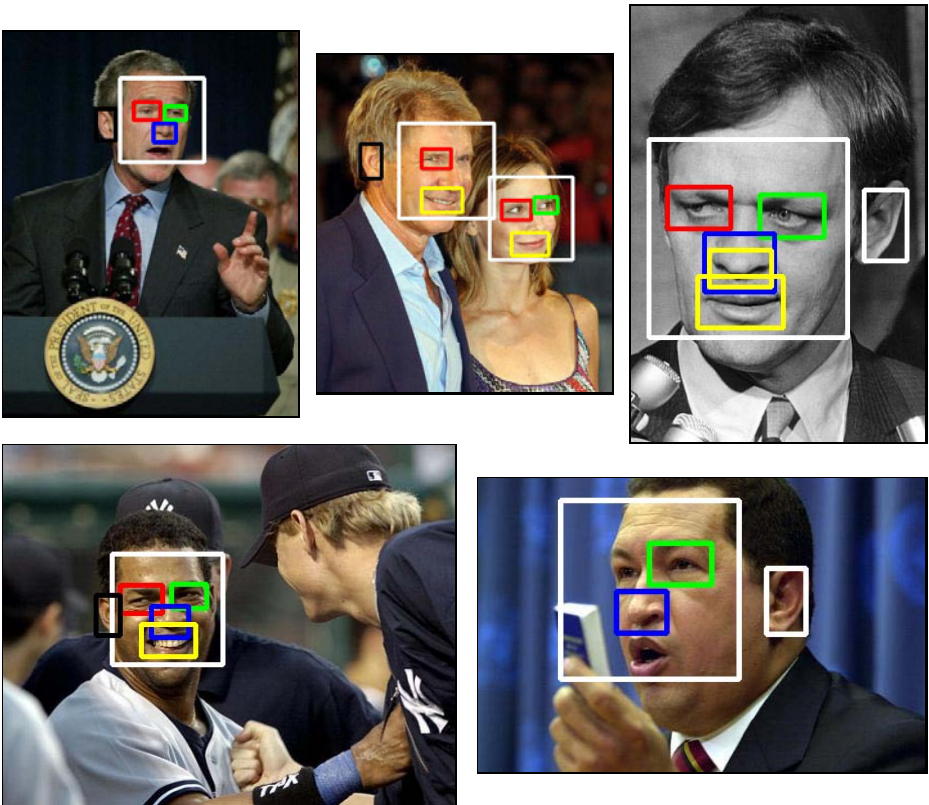


Fig. 5. Fddb detection samples with pose estimation based on facial features detection. The color code: red means left (in the image) eye detection, green means right (in the image) eye detection, blue means nose detection, yellow means mouth detection, black means left (in the image) ear detection, and white means right (in the image) ear detection

5 Conclusions

Observing the reliable classifiers trained by other researchers, we have used the Viola-Jones framework to train ear detectors. After the slow task of data gathering and training, they have been tested using the FERET database. Their respective detection results achieved have been presented suggesting a high detection rate. The specialized left and right ear detector performances evidences a detection rate larger than 92%, remarkably larger than the performance achieved by a general ear detector. These detectors are additionally reliable to be used in real-time applications employing standard webcams. These classifiers are therefore useful to any application requiring ear detection. For instance, we have applied the detector to the FDDB set to test the ability to suggest a lateral view. Other applications such as ear biometric systems, require an ear registration step that is now fast and simple.

We expect to explore further the combination of these classifiers with other facial feature detectors to improve face detection performance based on the combination of the evidence accumulation provided by inner facial feature detection. Such a detector would be more robust to slight rotations and occlusions.

Both classifiers reported in Figure 4b are now included in the OpenCV library. therefore other researchers can take them as baseline for comparison and improvement. In the next future, we will consider the addition of slightly rotated ear patterns to the positive set with the objective to analyze if a more sensitive classifier can be built.

References

1. Abaza, A., Hebert, C., Harrison, M.A.F.: Fast learning ear detection for real-time surveillance. In: Biometrics: Theory Applications and Systems, BTAS (2010)
2. Bradley, D.: Profile face detection (2003), <http://www.davidbradley.info> (last accessed July 15, 2011)
3. Castrillón, M., Déniz, O., Hernández, D., Lorenzo, J.: A comparison of face and facial feature detectors based on the viola jones general object detection framework. *Machine Vision and Applications* 22(3), 481–494 (2011)
4. Chang, K., Bowyer, K.W., Sarkar, S., Victor, B.: Comparison and combination of ear and face images in appearance-based biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 1160–1165 (2003)
5. Hewitt, R.: Seeing with OpenCV. a computer-vision library. *Servo*, 62–65 (January 2007)
6. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst (October 2007)
7. Intel. Open Source Computer Vision Library, v2.3 (July 2011), <http://opencv.willowgarage.com/wiki/> (last visited July 2011)
8. Islam, S.M.S., Bennamoun, M., Davies, R.: Fast and fully automatic ear detection using cascaded Adaboost. In: WACV (2008)
9. Islam, S.M.S., Bennamoun, M., Owens, R., Davies, R.: Biometric approaches of 2D-3D ear and face: A survey. *Advances in Computer And Information Sciences and Engineering*, 509–514 (2008)

10. Jain, A., Flynn, P., Ross, A.A. (eds.): Handbook of Biometrics. Springer, Heidelberg (2008)
11. Jain, V., Learned-Miller, E.: FDDB: A benchmark for face detection in unconstrained settings. Technical report, University of Massachusetts, Amherst (2010)
12. Kruppa, H., Castrillón-Santana, M., Schiele, B.: Fast and robust face finding via local context. In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), pp. 157–164 (October 2003)
13. Lienhart, R., Kuranov, A., Pisarevsky, V.: Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 297–304. Springer, Heidelberg (2003)
14. Lienhart, R., Liang, L., Kuranov, A.: A detector tree of boosted classifiers for real-time object detection and tracking. In: IEEE ICME 2003, pp. 277–280 (July 2003)
15. Lienhart, R., Maydt, J.: An extended set of Haar-like features for rapid object detection. In: IEEE ICIP 2002, vol. 1, pp. 900–903 (September 2002)
16. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(4), 607–626 (2009)
17. Jonathon Phillips, P., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face recognition algorithms. TR 6264, NISTIR (January 1999), <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00609311>
18. Reimondo, A.: Haar cascades repository (2007), <http://alereimondo.no-ip.org/OpenCV/34> (last visited April 2010)
19. Seo, N.: Tutorial: OpenCV haartraining (rapid object detection with a cascade of boosted classifiers based on haar-like features), <http://note.sonots.com/SciSoftware/haartraining.html> (last visited June 2010)
20. Vatahska, T., Bennewitz, M., Behnke, S.: Feature-based head pose estimation from images. In: Proceedings of IEEE-RAS 7th International Conference on Humanoid Robots, Humanoids (2007)
21. Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57(2), 151–173 (2004)

A Combined Strategy Using FMCDM for Textures Segmentation in Hemispherical Images from Forest Environments

P. Javier Herrera¹, Gonzalo Pajares², and María Guijarro²

¹ Dpto. Arquitectura Computadores y Automática, Facultad de Informática,
Universidad Complutense, 28040 Madrid, Spain
pjherrera@pdi.ucm.es

² Dpto. Ingeniería del Software e Inteligencia Artificial, Facultad de Informática,
Universidad Complutense, 28040 Madrid, Spain
{pajares, mgujarro}@fdi.ucm.es

Abstract. The research undertaken in this work comprises the design of a segmentation strategy to solve the stereoscopic correspondence problem for a specific kind of hemispherical images from forest environments. Images are obtained through an optical system based on fisheye lens. The aim consists on the identification of the textures belonging to tree trunks. This is carried out through a segmentation process which uses the combination of five single classical classifiers using the Multi-Criteria Decision Making method under Fuzzy logic paradigm. The combined proposal formulated in this research work is of unsupervised nature and can be applied to any type of forest environment, with the appropriate adaptations inherent to the segmentation process in accordance with the nature of the forest environment analyzed.

Keywords: Hemispherical forest images, segmentation, FMCDM, identification of textures, fisheye lens.

1 Introduction

The system based on the lens known as *fish-eye* are useful for inventories purposes because this optic system can recover 3D information in a large field-of-view around the camera. This is an important advantage because it allows one to image the trees in the 3D scene close to the system from the base to the top, unlike in systems equipped with conventional lenses where close objects are partially mapped [1].

Because the trees appear completely imaged, the stereoscopic system allows the calculation of distances from the device to significant points into the trees in the 3D scene, including diameters along the stem, heights and crown dimensions to be measured, as well as determining the position of the trees. These data may be used to obtain precise taper equations, leaf area or volume estimations [2].

The main contribution of this paper is the proposal for a strategy that solves one of the essential processes involved in stereo vision: segmentation of certain structures in the dual images of the stereoscopic pair. The strategy is designed according to the

type of images used and lighting conditions from forest environments. These refers to Scots pine forests (*Pinus sylvestris* L.) where images were obtained on sunny days and therefore exhibit highly variable intensity levels due to the illuminated areas. Due to the characteristics of this environment, the segmentation process is designed specifically according to this specific type of forest environment. This sets the trend for future research when analyzing other forest environments.

The segmentation process is approached from the point of view of isolating the trunks by excluding the textures that surround them (pine needles, ground, and sky). For this reason, we propose the use of the specific techniques of texture identification for the pine needles and of classification for the rest. This is carried out through the combination of five single classical classifiers using the Multi-Criteria Decision Making method under the Fuzzy logic paradigm (*FMCDM*).

This work is organized as follows. Section 2 describes the design of the segmentation process based on the *FMCDM* proposal. Section 3 describes the results obtained by using the combined approach, and comparing these results with those obtained by applying each individual strategy. Section 4 presents the conclusions and future work.

2 Combined Segmentation Strategy

In our approach, the interest is focused on the trunks of the trees because they contain the higher concentration of wood. These are our features of interest in which the later matching process is focused. Figure 1 displays two representative hemispherical images captured with a fisheye lens of the forest. As one can see there are three main groups of textures out of interest, such as grass in the soil, sky in the gaps and leaves of the trees. Hence, the first aim consists on the identification of the textures out the interest to be excluded during the matching process. This is carried out through a segmentation process which uses the combination of five classifiers under the *FMCDM* paradigm. The performance of combined classifiers has been reported as a promising approach against individual classifiers [3].

One might wonder why not to identify the textures belonging to the trunks. The response is simple. This kind of textures displays a high variability of tonalities depending on the orientation of the trunks with respect the sun. Therefore, there is not a unique type of texture (dark or illuminated trunks and even though alternatively in bits), as we can see in Figure 1. Observing the textures we can also see the following: *a)* the areas covered with leaves display high intensity variability in a pixel and the surrounding pixels in its neighborhood; *b)* on the contrary, the sky displays homogeneous areas; *c)* the grass in the soil also tend to fall on the category of homogeneous textures although with some variability coming from shades; *d)* the textures coming from the trunks are the most difficult as we said above; indeed due to the sun position, the angle of the incident rays from the sun produce strong shades in the part of the trunks in the opposite position of the projection, e.g. west part in the images of Figure 1(*a*); the trunks receiving the direct projection display a high degree of illumination, e.g. east part in the images of Figure 1(*a*); there are a lot of trunks where the shades produce different areas.

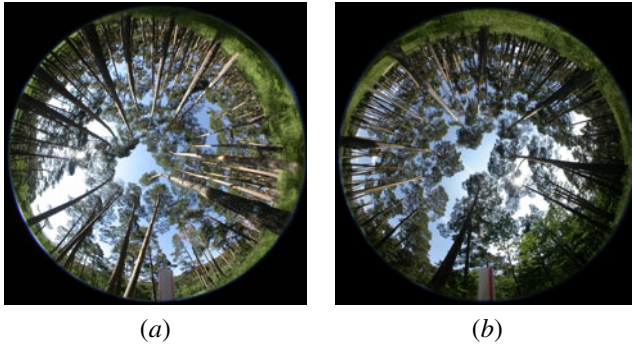


Fig. 1. Two representative hemispherical images

Based on the above, for identifying the textures coming from leaves, we use texture analysis techniques based on statistical measures that can cope with the high intensity variability. This is briefly explained in Section 2.1. Because of the homogeneity of grass and sky textures, we can use methods based on learning approaches as explained in section 2.2 where the combined proposal is also presented.

2.1 Identification of High Contrasted Textures

The textures produced by the leaves of the trees under analysis do not display spatial distributions of frequencies nor textured patterns; they are rather high contrasted areas without any spatial orientation. Hence, we have verified that the most appropriate texture descriptors are those capturing the high contrast, i.e. statistical second-order moments. One of the simplest is the *variance*. It is a measure of intensity contrast defined in our approach as in [4]. The criterion for identifying a high textured area is established by considering that it should have a value for the intensity contrast coefficient R , normalized in the range $[0, +1]$, greater than a threshold T_1 , set to 0.8 in this work after experimentation. This value is established taking into account that only the areas with large values should be considered, otherwise a high number of pixels could be identified as belonging to these kinds of textures because the images coming from outdoor environments (forests) display a lot of areas with different levels of contrast.

2.2 Identification of Homogeneous Textures: Combining Classifiers

As mentioned before, in our approach there are other two relevant textures that must be identified. They are specifically the sky and the grass. For a pixel belonging to one of such areas the R coefficient should be low because of its homogeneity. This is a previous criterion for identifying such areas, where the low concept is mapped assuming that R should be less than the previous threshold T_1 . Nevertheless, this is not sufficient because there are other different areas which are not sky or grass fulfilling this criterion.

Five single classical classifiers are used, three of them are of supervised nature and two of them of unsupervised nature, which form the basis for the design of the combined classification strategies proposed in this paper. Classic classifiers mean the

description, according to its original version. Individual classifier means the act of using a single classifier, to distinguish strategies that employ two or more classifiers, which we refer to as combined. Individual classifications are as follows [5]: *a*) Fuzzy Clustering (FC), *b*) Parametric Bayesian Classifier (PB), *c*) Parzen window (PZ), *d*) Generalized Lloyd algorithm (GL), and *e*) Self-Organizing Maps (SOM). The choice of these classifiers is based on its proven effectiveness at individual in various fields of application, including image classification.

As mentioned before, the combination of classifiers improves the results. We choose one of many possible options for the combination of the five individual classifiers, opting for the Multi-Criteria Decision Making method under Fuzzy logic paradigm, because of nature fuzzy, which allows for some flexibility regarding the images used in the experiments carried out in this work.

Despite the implicit supervised nature in three of the five individual classifiers, the combined proposal formulated in this research work is of unsupervised nature. This is achieved through the design of the proposed strategy, which allows distribution of samples into classes automatically, that is, carrying out a partition along with the validation process for that partition. For this reason, it is necessary first to determine the process for obtaining partition and secondly, establishing the criteria for validation.

Any classification process in general and in particular the identification of textures in natural images has associated two main phases: *training* and *decision*. We refer to the first phase as *learning* phase also, by identifying both concepts in the literature.

2.2.1 Training and Decision Phases of the Individual Classifiers

Now we give a brief description of the five individual classification methods involved in the design of combined classifier used in the process of segmentation in this work.

The aim of FC technique is to estimate the centers of the classes and grades of each sample belonging to each class. The decision is the process by which a new sample \mathbf{x}_s whose membership to some class is unknown so far, must be identified as belonging to a class w_j available. This classifier has been widely used in literature, whose description can be found in [5,6], among others.

PB method has traditionally been identified within the unsupervised classification techniques [7]. Given a generic training sample $\mathbf{x} \in \mathfrak{R}^q$ with q components, the goal is to estimate the membership probabilities to each class w_j , i.e. $P(w_j | \mathbf{x})$. This technique assumes that you know the density function of conditional probability for each class, resulting in unknown parameters or statistical involved in this task. A widespread practice, adopted in this paper, is to assume that the shape of these functions follows the law of Gaussian or Normal distribution.

In PZ process, as in the case PB method, the goal remains the calculation of membership probabilities of sample \mathbf{x} to each class w_j , that is $P(w_j | \mathbf{x})$. In this case there are no parameters to be estimated, except the probability density function [6].

GL was originally proposed by [8] and later generalized for vector quantization by [9]. The method used in this work is a modified version of the original GL and is known as *competitive learning algorithm* in neural network literature. The objective of the decision phase is to classify a new sample \mathbf{x}_s in any existing class w_j . To this end, centres (weights) \mathbf{c}_j stored during the training phase are recovered from *Knowledge Base (KB)*, determining the proximity of the sample to all class centres. The

proximity is established based on a minimum distance criterion. This distance should be the same as that used in the training phase, so the Euclidean.

SOM is one of the most popular neural networks used for reducing the dimensionality of the data. It is described in [5]. The implementation of the SOM algorithm for natural textures images classification requires the definition of the input patterns and connection weights and the number of neurons in the Kohonen layer. Thus, the input vectors are the vectors \mathbf{x} that characterize the spectral components of the pixels as RGB colour model adopted in our case. Therefore, the number of neurons in the input layer is three, corresponding to each of the three components R, G and B used. The number of neurons in the output layer is determined by the number of existing classes.

2.2.2 Quality of the Partition

To estimate the parameters derived from the learning process, the five methods briefly described above require the distribution in c classes, w_1, w_2, \dots, w_c of the n samples contained in $X = \{x_1, x_2, \dots, x_n\} \in \mathfrak{R}^q$, i.e. the partition of the samples. This distribution can be done in two ways, one is manually under supervision of an expert, and the other is automatically unsupervised. The latter is adopted here with the aim of achieving the automation of the learning process, which is one of the objectives of the work.

How many partitions of the samples unsupervised can be considered valid? Validation of the partitions has been a frequent topic in literature as can be inferred from studies by [10] and associated references. It is commonly accepted that as more similar are the samples together in the same class and more differences among samples from different classes, the better the partition. Therefore, the objective is to obtain a partition that best meets the two previous premises.

The classifier in which we rely to make the initial partition into classes is FC, which estimates the membership degree of the samples belonging to classes. To this end, we have considered different criterion functions that consist of scalar measurements to validate the initial partition, which are the *Partition Coefficient (PC)*, *Partition Entropy (PE)* and *Xie-Beni index (XB)* [11].

On the other hand, there are still two issues related to the criterion functions useful that can be exploited properly. First, during the study of the combined classifier, one of the problems is to determine the relative importance of each individual classifier in the combination, i.e. which of them has a better or worse behavior. In this work we have designed a procedure to infer the behavior of individual classifiers based on the values provided by the criterion functions. Second, the individual classifiers studied so far, except PB and SOM classifiers, are supervised in nature, either by definition or approach. Well, thanks to the behavior of criterion functions, we can automate the design process and gain unsupervised.

There are other functions based on compaction and separation of classes, such as *Fukuyama and Sugeno* or *Kwon* among others, and can be found in [11], but after various experiments, their behavior does not introduce any significant contribution with respect to those mentioned above, so they have not been considered.

PC criterion function is monotonically increasing, while PE and XB are monotonically decreasing in the three cases according to the number of classes. The focus is on obtaining a criterion for determining under what conditions maximum or minimum can be considered valid partition.

Given the complexity of the forestry images treated, it is sometimes very difficult to determine exactly the total number of textures. If you set a strict validation criterion and generate too many classes (in our case more than six), it is very difficult to determine which of them cover completely the existing textures in the images and in particular those related to tree trunks, textures that are of priority interest. That is why in this type of images, it should not partition criteria are too strict, or that the criterion functions have too much variation to determine the number of classes needed for a better classification of the trunks.

PC and PE indexes have been proven more stable in tests, while XB varies greatly depending on the type of images and the pixels selected to form the initial partition. On the other hand, instead of trying to determine what criterion function performs better than the others to validate the initial partition generated, we have tried to combine the criteria most stable, i.e. PC and PE, finding threshold values so when the relative variation is below, the partition is considered valid.

2.2.3 Fuzzy Multi-criteria Decision Making

This method performs the pixel-level combination, scheduled for this reason as *local* in nature. The features are, therefore, pixels. The three spectral components of these features in the RGB color model are the properties used. The combination of individual classifiers is performed during the decision phase.

Under the local approach is proposed a method that uses five individual classifiers mentioned previously, i.e. FC, PB, PZ, GL, and SOM. The following describes the combined method, giving details of same in regard to the training phase and decision.

Figure 2 shows the scheme of unsupervised classifier based on the FMCDM paradigm. As can be seen, the procedure works in the above mentioned two phases of training and decision.

The training process begins with processing the training patterns or samples available, i.e. the inputs to the system. The input pattern is the same for all classifiers. Initially, we assume the existence of a single class and all samples within that category, so $c = 1$. Under this assumption, we establish a partition of the samples in the only class available at this moment. The partition in a single class is considered invalid by definition, since in the images available this never occurs. For this reason, we try a new partition with $c = 2$. After which, we evaluate if the new partition is valid or not.

The validation process is carried out through a combination of PC and PE criteria, as we explained in section 2.2.2. If the partition is not valid, according to the above criterion, the number of classes c is incremented by one, proceeding again to repeat the previous process to get the validation of the partition. The distribution of samples in classes, once known the number of members will be carried out by means of pseudo-random process described in [12]. This is the basic process which gives this design its unsupervised nature [6]. By contrast, when the partition is considered valid, also according to the same above criterion, the five individual classifiers perform their respective training processes to carry out the estimation of its parameters. After the respective training processes, the parameters estimated or learned by each of these classifiers are stored in the KB and shall be available for later retrieval and use during the decision phase, where the combination of classifiers is produced.

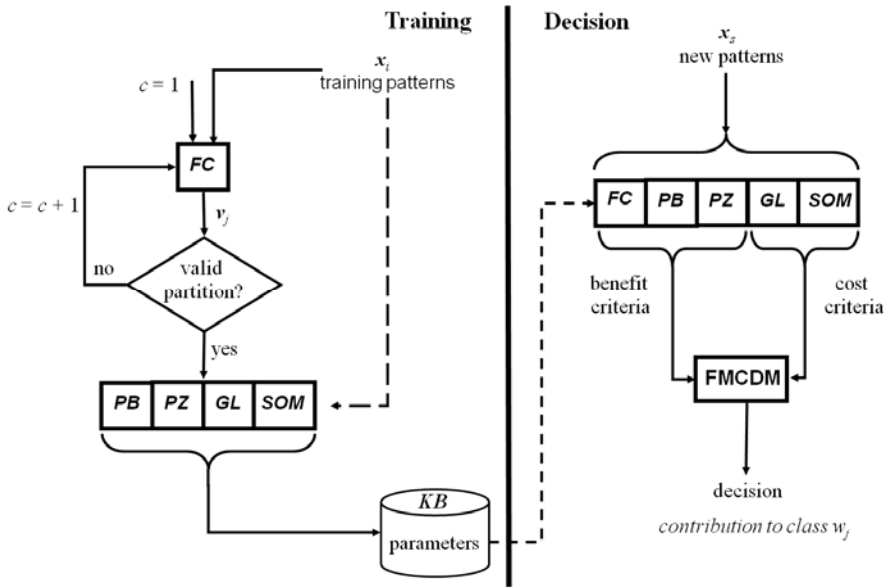


Fig. 2. Design of the combined unsupervised classifier: training and decision phases

As shown in Figure 2, given a new sample or pattern x_s , the problem that arises now is to make a decision on the classification of the sample in each of the classes available and previously established during the training.

Each individual classifier makes its own decision on the classification of the sample. Thus, FC provides the degrees of membership of that sample to each of the classes; both PB and PZ generate membership probabilities of that sample to each of the classes; GL and SOM provide sample distances to the centres of classes. The first three take the decision based on the maximum values of their outputs, while the two latter do so in terms of minimum values.

To apply the FMCDM paradigm is necessary to take all of these output values (degrees of membership, probabilities and distances) to merge following the previous steps and based on the work of [13,14,15]. We start from that the number of classes c has been estimated during the training phase. From the point of view of decision theory, the problem is to determine which class belongs x_s to. From the point of view FMCDM theory, the choice of a class is equivalent to choosing an alternative. Therefore, classes are identified as alternatives. The criteria for choosing an alternative in the FMCDM paradigm are determined by the outputs provided by individual classifiers. There are two types of criteria, namely: *benefit* and *cost*. As explained above, FC, PB and PZ classifiers make decisions based on the maximum values of their outputs and GL and SOM do according to the minimum.

A triangular fuzzy number is defined as (a_1, a_2, a_3) , where a_1 is the minimum possible value, a_2 is the value and a_3 is the best value possible. Instead of forming a triangular number ordering the outputs of three classifiers, we created a shortlist for each classifier, where a_2 corresponds exactly to the output of the classifier and a_1 and a_3 are generated respectively adding and subtracting random values according to the

range of values for each criterion. The calculation of bounds for the random values and obtaining the relative importance of each criterion, i.e. the calculation of specific weights associated are adjusted during the experimentation phase by the cross validation method [6].

The last step for choosing the best alternative, i.e. the most appropriate class w_j for the sample input x_s , which in our case are the values of the three spectral components in RGB colour model of the pixels in the forest images used. Since in our model alternatives and classes are equivalent, the best class w_j for x_s corresponds to the alternative that provides the maximum value according to the proximity coefficient calculated during the application of FMCDM [13].

Figure 3 shows from left to right, an original left image used as a reference, the result obtained by the output of the process proposed (combining classifiers and variance), and the two classes in this image have been considered as trunks.

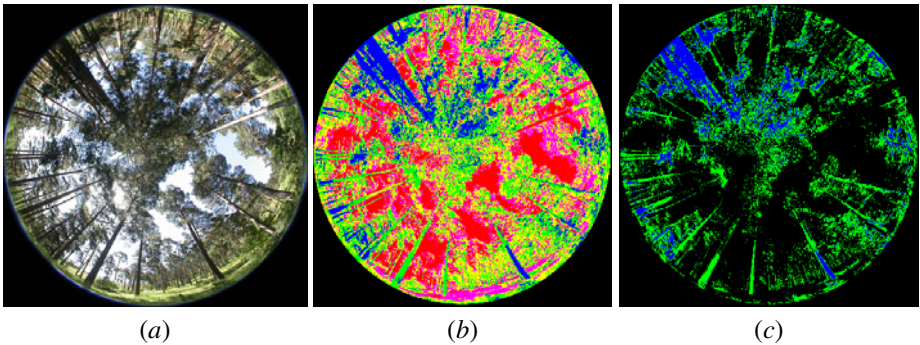


Fig. 3. (a) Original left image; (b) segmented left image where classes belonging leaves (pink and yellow), grass (pink), sky (red) and trunks (blue and green) are identified; (c) two classes where trunks are identified

Once the image segmentation process is finished, we have identified pixels belonging to tree trunks and three types of textures without interest. In the future, pixels belonging to textures without interest will be discarded during the next stereovision matching process. Hence, we only apply the stereovision matching process to the pixels that do not belong to any of these textures.

3 Results

The camera is equipped with a Nikon FC-E8 fisheye lens, with an angle of 183° . The valid color images in the circle contain 6586205 pixels. The tests have been carried out with twenty pairs of stereo images from *Pinus Sylvestris* L. forests obtained on sunny days. We use four of them to find the best possible configuration of the specific weights for the individual classifiers used in the FMCDM through the cross validation method [6], and the calculation of specific weights associated to triangular fuzzy numbers. At a second stage, we apply the five individual classifiers and the combined proposal for the remainder sixteen stereo pairs.

On each classifier, it is necessary to synthesize and interpret results with the outputs of other classifiers for comparative purposes. This has been realized by means of ground truth images obtained manually.

Table 1 displays the averaged percentage of errors and percentage of trunk pixels successfully classified obtained through five individual classifiers and the combined proposal. As one can see FC obtains the better results on average percentage of error followed by our proposal. However, according to the percentage of trunk pixels successfully classified obtained with each method, the better results are obtained with FMCDM.

Table 1. Averaged percentage of errors and percentage of trunk pixels successfully classified obtained through the combined method proposed against the individual classifiers

<i>Classifiers</i>	<i>% (e)</i>	<i>% (t)</i>
FC	19.82	62.50
PB	24.12	60.93
PZ	33.54	60.13
GL	23.93	61.72
SOM	29.30	60.92
FMCDM	22.78	65.61

4 Conclusions and Future Work

This paper presents an automatic strategy of segmentation for identifying textures belonging to tree trunks from hemispherical stereo images captured with fisheye lenses. The interest is focused on the trunks of the trees because they contain the higher concentration of wood for inventories purposes.

Five singles classical classifiers are used which form the basis for the design of the combined classification strategy proposed in this paper. This is carried out through a segmentation process which uses the combination of classifiers using the Fuzzy Multi-Criteria Decision Making method. While others individual classifiers might have chosen as a different combined strategy, the combination of these in relation to the improvement of the results according to the set of images used shows its promising possibilities. The computational cost of this proposal is acceptable although it would be desirable a lower cost. This is proposed as future work. The proposed strategy based on segmentation process can be favorably compared from the perspective of the automation of the process and we suggest it can be applied to any type of forest environment, with the appropriate adaptations inherent to the segmentation process in accordance with the nature of the forest environment analyzed.

Acknowledgments. The authors wish to acknowledge to the Council of Education of the Autonomous Community of Madrid and the Social European Fund for the research contract with the first author. Also to the Spanish Forest Research Centre (CIFOR-INIA) for their support and the imaged material supplied. This paper has been funded under project DPI2009-14552-C02-01 from the Ministerio de Educación y Ciencia of Spain within the Plan Nacional of I+D+i.

References

1. Abraham, S., Förstner, W.: Fish-Eye-Stereo Calibration and Epipolar Rectification. *Photogram. Remote Sens.* 59, 278–288 (2005)
2. Montes, F., Ledo, A., Rubio, A., Pita, P., Cañellas, I.: Use of Stereoscopic Hemispherical Images for Forest Inventories. In: *Proc. Int. Scientific Conference Forest, Wildlife and Wood Sciences for Society Development*, Faculty of Forestry and Wood Sciences, Czech University of Life Sciences, Prague (2009)
3. Kuncheva, L.I.: *Combining Pattern Classifiers: Methods and Algorithms*. Wiley (2004)
4. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*, 3rd edn. Prentice Hall, New Jersey (2008)
5. Pajares, G., Cruz, J.M.: *Visión por Computador: Imágenes Digitales y Aplicaciones*, 2nd edn., Ra-Ma, Madrid (2007)
6. Duda, R.O., Hart, P.E., Stork, D.S.: *Pattern Classification*. Wiley & Sons, New York (2001)
7. Escudero, L.F.: *Reconocimiento de patrones*. Paraninfo, Madrid (1977)
8. Lloyd, S.P.: *Least Squares Quantization in PCM's*. Bell Telephone Laboratories Paper. Murray Hill, New Jersey (1957)
9. Linde, Y., Buzo, A., Gray, R.M.: An Algorithm for Vector Quantization Design. *IEEE Trans. Communications* 28(1), 84–95 (1980)
10. Volkovich, Z., Barzily, Z., Morozensky, L.: A Statistical Model of Cluster Stability. *Pattern Recognition* 41(7), 2174–2188 (2008)
11. Kim, D.W., Lee, K.H., Lee, D.: Fuzzy Cluster Validation Index Based on Inter-Cluster Proximity. *Pattern Recognition Letters* 24, 2561–2574 (2003)
12. Balasko, B., Abonyi, J., Feil, B.: *Fuzzy Clustering and Data Analysis Toolbox for Use with Matlab*, Veszprem University, Hungary (2006)
13. Wang, W., Fenton, N.: Risk and Confidence Analysis for Fuzzy Multicriteria Decision Making. *Knowledge Based Systems* 19, 430–437 (2006)
14. Gu, X., Zhu, Q.: Fuzzy Multi-Attribute Decision-Making Method Based on Eigenvector of Fuzzy Attribute Evaluation Space. *Decision Support Systems* 41, 400–410 (2006)
15. Chen, C.T.: Extensions of the TOPSIS for Group Decision-Making under Fuzzy Environment. *Fuzzy Sets and Systems* 114, 1–9 (2000)

Getting NDVI Spectral Bands from a Single Standard RGB Digital Camera: A Methodological Approach

Gilles Rabatel¹, Nathalie Gorretta¹, and Sylvain Labbé²

¹ Cemagref, UMR ITAP, 361 rue Jean-François Breton, BP 5095, 34196 Montpellier Cedex 5, France

² Cemagref, UMR TETIS, F-34093 Montpellier, France

Abstract. Multispectral images including red and near-infrared bands have proved their efficiency for vegetation-soil discrimination and agricultural monitoring in remote sensing applications. But they remain rarely used in ground and UAV imagery, due to a limited availability of adequate 2D imaging devices. In this paper, a generic methodology is proposed to obtain simultaneously the near-infrared and red bands from a standard RGB camera, after having removed the near-infrared blocking filter inside. This method has been applied with two new generation SLR cameras (Canon 500D and Sigma SD14). NDVI values obtained from these devices have been compared with reference values for a set of soil and vegetation luminance spectra. The quality of the results shows that NDVI bands can now be acquired with high spatial resolution 2D imaging devices, opening new opportunities for crop monitoring applications.

Keywords: NDVI, aerial imaging, multispectral, near-infrared band.

1 Introduction

The Normalized Difference Vegetation Index, or NDVI, introduced in the early seventies by [1], remains today a very popular tool in the remote sensing community dealing with agricultural monitoring. This is mainly due to its remarkable ability to discriminate vegetation from other material in multispectral satellite images. Green vegetation is characterized by a high reflectance in the near-infrared domain (typically 50 to 80%), which contrasts with a very low reflectance in the red wavelengths, due to chlorophyll absorption. Let us call R and NIR the digital counts obtained through the red and the near infrared bands of a multispectral sensor. The NDVI, expressed as:

$$\text{NDVI} = \frac{\text{NIR} - \text{R}}{\text{NIR} + \text{R}} \quad (1)$$

is a scalar value in the range [-1, 1]. The higher is this value, the higher is the probability that it corresponds to green vegetation. By extension, numerous

attempts have been made to directly link NDVI or other derived indexes based on R and NIR to agronomical indices such as biomass and LAI (Leaf Area Index) [2], [3], [4].

The popularity of NDVI in remote sensing has been widely supported by the availability of R and NIR channels on most satellite line-scanning sensors (Landsat, SPOT, etc.). Unfortunately, it is not the case for crop monitoring applications at a lower spatial scale: most vision systems embedded on ground vehicles or UAV (Unmanned Aerial Vehicle), which require 2D sensors, are based on standard colour cameras, leading to robustness issues in vegetation detection.

In most colour cameras, the RGB channels are obtained by setting a mosaic of microfilters (Bayer matrix) directly on the CCD (or CMOS) sensor. This solution requires a very large production scale, and the development of specific near-infrared bayer matrices for the vegetation monitoring market cannot be envisaged. Therefore, for agricultural applications, some camera manufacturers propose multi-CCD devices including a near infrared channel, e.g. the MS-4100 (Geospatial Systems Inc., West Henrietta, NY, USA), or the AD-080 (JAI Ltd, Yokohama, Japan). But their price is often prohibitive. Moreover, their spatial resolution is hardly sufficient for UAV image acquisition.

To face this situation, some camera end users requiring a near infrared channel have developed alternative solutions around standard Bayer matrix RGB cameras, taking benefit of an undesirable property of their silicium sensing array: because the colour filters in the Bayer matrix have a filtering action limited to the visible domain, the camera manufacturers are constrained to add a near-infrared blocking filter to match natural colorimetry requirements. By removing this additional filter, we obtain a modified camera sensitive to near infrared wavelengths. A first possibility to get separate R and NIR bands is thus to use simultaneously a standard and a modified colour camera, the second one being equipped with a near infrared pass-band filter [5]. However, important issues arise concerning the pixel alignment of the two images obtained [6].

Another interesting approach is to use a single modified camera associated with a low-pass filter, and to built the near infrared band as a specific linear combination of the three resulting channels. This concept can be illustrated as following:

Let us assume we have an ideal modified RGB camera, where the three R,G,B channels deliver digital counts respectively equal to $R+NIR$, $G+ NIR$, $B+NIR$. If we add a low-pass filter in front of the lens that blocks the blue wavelengths, then we get only the NIR component on the blue channel. This component can be subtracted from the other channel digital counts, leading finally to R, G and NIR components. In the real world, the sensitivity of each channel cannot be modeled so simply, and a specific study is necessary to determine the required low-pass filter and the linear combination for a given camera. Until now, this approach has been used by the company Tetracam (Tetracam Inc. Chatsworth, CA, USA) in its agricultural cameras (ADC series). However, their spatial resolution (~ 3 Mpixels) does not meet the present standard of RGB still cameras (more than 10 Mpixels).

The purpose of the present study is to formalize this last approach and to show how it can be extended to the newest generation of commercial RGB imaging sensors, in order to combine robust spectral information on vegetation with very high spatial resolution. A generic methodology is proposed to determine the optimal low-pass filter and linear combination for virtually any modified RGB camera, provided its sensitivity curves are known. The results obtained with two commercial imaging devices representative of the recent evolutions in sensor technology, the Canon 500D (Canon, Tokyo, Japan) and the Sigma SD14 (Sigma, Kawasaki, Japan) are then presented.

2 The BSOP Theoretical Approach

The basic idea developed here is to simulate a desired spectral sensitivity (further referred as target sensitivity) by a linear combination of the real spectral sensitivities available for a given sensor associated with a low-pass filter. In the following, we will express formally this linear combination, and then propose a method to determine the optimal low-pass filter to be associated with the sensor.

2.1 Band Simulation by Orthogonal Projection (BSOP)

Formally, the spectral sensitivity of a sensor channel (or band) can be characterised by a function $v(\lambda)$ of the wavelength λ . It will generate, for a given irradiance $e(\lambda)$, a digital count:

$$DC = \int_0^{\infty} e(\lambda).v(\lambda).d\lambda \quad (2)$$

For practical reasons, we will consider in the following a limited range of wavelengths and a limited spectral resolution, allowing us to consider any spectral function as a vector in a spectral space of dimension n , leading to the discrete expression:

$$DC = \sum_{i=1}^n e(\lambda_i).v(\lambda_i) = E.V \quad (3)$$

where E and V are vectors of dimension n and $E.V$ is their scalar product.

According to this formalism, let us call V_1, \dots, V_p the actual sensitivities of the p bands of an imaging sensor. By linear combination of V_1, \dots, V_p , we can simulate any virtual sensitivity V belonging to the subspace generated by the vectorial base (V_1, \dots, V_p) . Now, let us consider a given target sensitivity V_t that we want to simulate. In the general case, V_t will not belong to this subspace, and the better approximation of V_t will be its orthogonal projection on the (V_1, \dots, V_p) subspace (Fig. 1).

Let us call $B = [V_1 \dots V_p]$ the (n, p) matrix whose columns are the V_1, \dots, V_p vectors. The orthogonal projection of V_t can be expressed as:

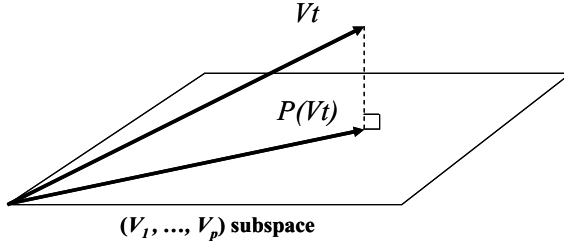


Fig. 1. Illustration of the orthogonal projection of the target sensitivity V_t

$$P(V_t) = B.(B^T B)^{-1}.B^T.V_t \tag{4}$$

The coordinates of $P(V_t)$ in the base (V_1, \dots, V_p) , i.e. the coefficients of the linear combination giving $P(V_t)$ from V_1, \dots, V_p , are given by the vector of dimension p :

$$C = B.(B^T B)^{-1}.B^T.V_t \tag{5}$$

According to (4) and (5):

$$P(V_t) = B.C = [V_1 \dots V_p].C = c_1.V_1 + \dots + c_p.V_p \tag{6}$$

2.2 Optimal Low-Pass Filter

Whatever is the set of spectral sensitivities V_1, \dots, V_p and the target sensitivity V_t , the projection vector $P(V_t)$ defined above will always exist, but this does not guarantee a satisfactory result. We have still to verify that $P(V_t)$ is close to the target vector V_t . A commonly used measure of the similarity of spectral data is the *SAM*, or Spectral Angle Mapper [7]. This measure evaluates the angle between two vectors v_1 and v_2 of euclidian norms $\|v_1\|$ and $\|v_2\|$ as:

$$SAM(v_1, v_2) = \text{acos}\left(\frac{v_1.v_2}{\|v_1\|.\|v_2\|}\right) \tag{7}$$

In our case, the value $SAM(V_t, P(V_t))$ should be ideally equal to zero, meaning that V_t belongs to the subspace (V_1, \dots, V_p) . In the real case, where the initial spectral sensitivities of the different channels are a specification of the sensor, there is no a priori reason to meet this requirement. Let us call (V_1^*, \dots, V_p^*) these initial sensitivities. Our only degree of freedom to improve the situation is to associate with the sensor an optical filter (e.g. by setting it in front of the lens) with a spectral transmittance F , leading to a modified set of sensitivities:

$$(V_1, \dots, V_p) = (F.V_1^*, \dots, F.V_p^*) \tag{8}$$

Our objective is thus to determine the optical filter F that will minimize the SAM between the target V_t and its projection $P(V_t)$, according to equations (4) and (8). A first possible method would be to directly search for an optimal filter F in the spectral space by minimisation techniques. Such a method would be rather complex to develop, but overall it could lead to non-realisable optical filters, unless constraints of positivity and smoothness are introduced. At the present stage, according to the first considerations described in introduction, we have chosen to reduce the complexity of the problem by considering only simple low-pass filters as F candidates. A main advantage is the further possibility to implement the solution with on-the-shelf gelatine or glass filters.

Let us define a candidate low-pass filter by its cutting wavelength λ_c :

$$F_{\lambda_c}(\lambda) = 1 \quad \text{if } \lambda > \lambda_c; \quad F_{\lambda_c}(\lambda) = 0 \quad \text{otherwise}; \tag{9}$$

and let us consider k target sensitivities V_{t1}, \dots, V_{tk} .

Then for a given wavelength λ_c we can compute:

- the subspace matrix $B_{\lambda_c} = [V_{1\lambda_c}, \dots, V_{p\lambda_c}] = [V_1 \cdot F_{\lambda_c}, \dots, V_p \cdot F_{\lambda_c}]$
- the projected vectors $P_{\lambda_c}(V_{t1}), \dots, P_{\lambda_c}(V_{tk})$, according to equation (4)
- a global cost function taking into account the similarity between every target and its projection:

$$R(\lambda_c) = SAM(V_{t1}, P_{\lambda_c}(V_{t1})) + \dots + SAM(V_{tk}, P_{\lambda_c}(V_{tk})) \tag{10}$$

The optimal low-pass filter F_{λ_c} will be the one that minimise $R(\lambda_c)$.

2.3 Renormalisation of the Projected Vectors

The SAM criterion above has been used to determine a set of vectors $P_{\lambda_c}(V_{t1}), \dots, P_{\lambda_c}(V_{tk})$ matching as well as possible an initial set V_{t1}, \dots, V_{tk} of target sensitivities, in terms of spectral shape. Another important point, if these vectors are devoted to the computation of agricultural indices like NDVI, is that they provide digital counts as close as possible to the original ones. Though this can obviously not be obtained for every irradiance spectrum, an approximate solution is to ensure that the projected vector and the target vector have the same $L1$ -norm¹.

Therefore, the following renormalisation is finally applied to each projected vector:

$$\forall i \in [1, k], \quad P_N(V_{t_i}) = P(V_{t_i}) \cdot \frac{\|V_{t_i}\|_{L1}}{\|P(V_{t_i})\|_{L1}} \tag{11}$$

¹ The $L1$ -norm of a vector is defined as the sum of the absolute values of its components. If its components are all positive, the $L1$ -norm of a sensitivity vector is equal to its scalar product with a flat spectrum $E_f = (1, \dots, 1)$, i.e. is equal to its digital count for an irradiance E_f .

3 Material and Methods

3.1 Camera Sensitivity Measurement

Two commercial still cameras have been selected for the BSOP method assessment: the Canon 500D, representative of recent high resolution Bayer matrix sensors (15 Mpixels), and the Sigma SD14, for the original technology of its Foveon X3 sensor. The Foveon sensor is made of three separate layers of photodetectors. Since silicon absorbs different wavelengths at different depths, each layer captures a different color. No Bayer matrix is required, leading to a better spatial resolution. In counterpart, there is less control on the spectral sensitivity curve of each channel.

Each camera has been opened and its NIR blocking filter removed (the filter is removable in the Sigma, but not in the Canon, leading to a more delicate operation). The spectral sensitivity of each modified camera has then been measured in the range 440-990 nm, with 10 nm steps. For this purpose, the tunable monochromatic light source of a laboratory spectrometer (V-570, Jasco Inc, Easton, USA) have been remoted in front of the camera lens via an optical fiber, at a 30 cm distance, in a room with no ambient light. Pictures have been taken in raw format for each wavelength in the following conditions:

Focal length: 50 mm; Sensibility: 100 ISO. Integration time: 5 s

The average level of the light source image has then been collected for each channel in the raw camera pictures, using home-made image analysis software.

3.2 Definition of the Target Sensitivities

Two target sensitivities V_{t1} , V_{t2} corresponding respectively to the red and near-infrared bands have been considered. Because no standardized sensitivity data have been found in the literature (some authors use the bands TM3 and TM4 of Landsat, other simply use 660 nm and 760 nm wavelengths), the following procedure has been used:

- the red band has been derived from the XYZ colorimetric data of CIE 1931 (source: <http://www.cvrl.org/database/text/cmfs/ciexyz31.htm>)
- the near-infrared band has been computed by a 160 nm shift of the red band, leading to a bandwidth 760-830 nm at mid-height.

3.3 BSOP Computation

In order to be used further on real hyperspectral data (see section 3.4), all the data defined above (sections 3.1 and 3.2) have been resampled according to the spectral resolution of an hyperspectral camera model Hypspec VNIR-1600 (Norsk Elektro Optikk A/S, Norway), i.e. 160 spectral bands from 415 to 993 nm.

According to the notations defined in section 2, this has led to two sets of vectors V_1^* , V_2^* , V_3^* (one for each type of camera) and two vectors V_{t1} , V_{t2} of dimension 160. All computations described in section 2 have been made with Matlab

7 (The MathWorks, Natick, MA, USA). Once the optimal cutting wavelength has been determined, the closest existing gelatine Wratten filter has been chosen (see 4.1). Its real transmittance curve has been measured with the Jasco V-570 spectrometer, and resampled according to the Hypsux 160 bands, leading to a filter vector F_w . Finally, for each type of still camera, the projected vectors $P_N(V_{t1})$ and $P_N(V_{t2})$ on the subspace $(F_w \cdot V_1^*, F_w \cdot V_2^*, F_w \cdot V_2^*)$ have been computed and renormalised according to (11).

3.4 NDVI Simulation on Field Hyperspectral Images

Real hyperspectral images of wheat durum have been acquired in experimental fields (INRA, Domaine de Melgueil, France) in march 2011, using the Hypsux VNIR-1600 camera. The camera was set on a motorized translation rail one meter above the ground (see [8] for more details). The scenes include wheat durum at early stage, as well as various types of dicotyledonous and monocotyledon weeds. A total of 2210 luminance spectra have then been collected in the hyperspectral images by manual area selections, including wheat, weeds and soil categories.

For every collected spectrum S , the digital counts R and NIR corresponding to red and near infrared target sensitivities V_{t1} , V_{t2} have been computed as $R = S \cdot V_{t1}$ and $NIR = S \cdot V_{t2}$, according to equation (3). A reference NDVI value has then been computed using this R and NIR values. The same operation has been made using $P_N(V_{t1})$ and $P_N(V_{t1})$ instead of V_{t1} and V_{t2} to compute an estimated NDVI for each type of still camera.

4 Results

4.1 Band Simulation

Fig.2 shows the spectral sensitivities V_1^* , V_2^* , V_3^* that have been measured respectively for the Canon and Sigma cameras without their NIR blocking filter.

In Fig.3, the BSOP quality curves $R(\lambda_c)$ according to equation (10) have been reported. We can notice that in both cases, $R(\lambda_c)$ is undefined for $\lambda_c > 728$ nm, because the projection of V_{t1} (red band) becomes null. The best BSOP quality is obtained for $\lambda_c \approx 600$ nm, and is better for the Canon camera. According to this result, a standard Wratten filter Kodak $n^\circ 25$ (reference: 149 7072) with a cutting wavelength of 600 nm has been selected, and its actual transmittance F_w measured with the Jasco spectrometer for further simulations.

The results of BSOP are given in Fig.4. The simulated near infrared band obtained with the Canon camera appears more regular than the one obtained with the Sigma camera, confirming the better results obtained with $R(\lambda_c)$.

4.2 NDVI Computation

Fig. 5 shows the NDVI values computed with the simulated bands of Fig.4, versus the reference NDVI values computed with the target sensitivities V_{t1}, V_{t2} , for the 2210 luminance spectra that have been collected (see section 3.4).

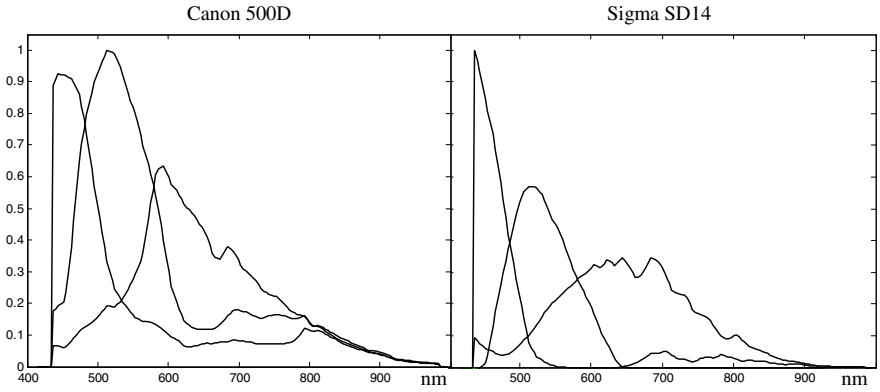


Fig. 2. Spectral sensitivities of the Canon 500D and Sigma SD14 without NIR blocking filter (sensitivities have been rescaled for a maximum value equal to unity)

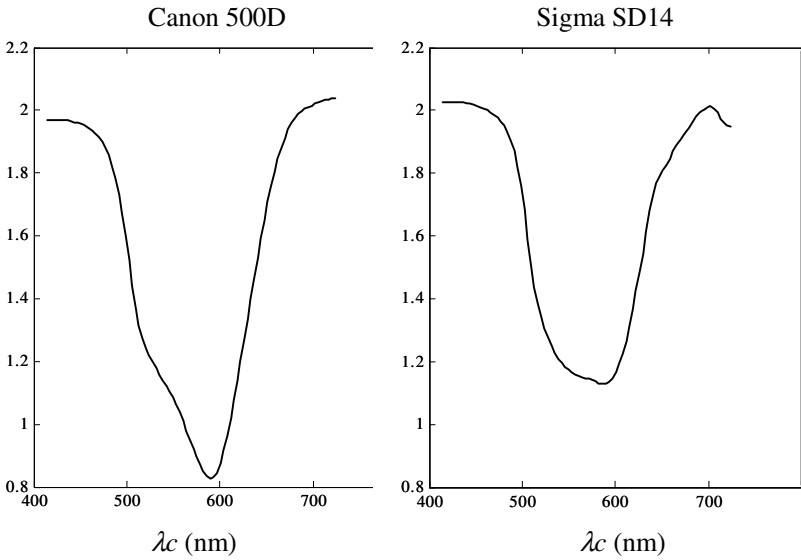


Fig. 3. BSOP quality $R(\lambda_c)$ of Canon and Sigma cameras

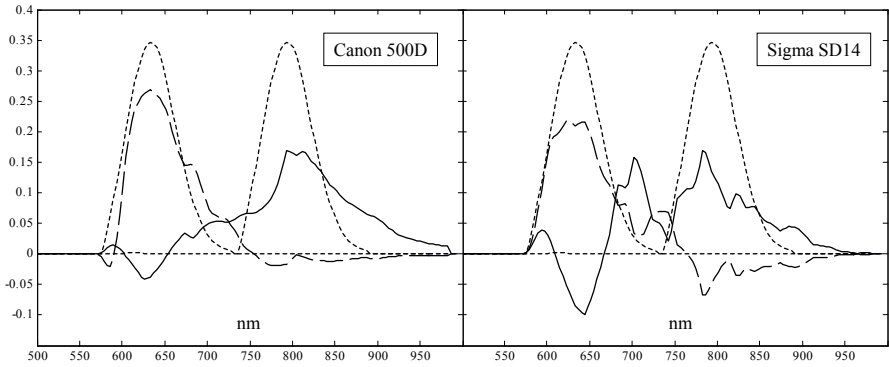


Fig. 4. BSOP bands for Canon and Sigma cameras with Wratten filter n° 25. Dotted lines: target sensitivities V_{t1}, V_{t2} . Dash line: BSOP red band; Solid line: BSOP near-infrared band.

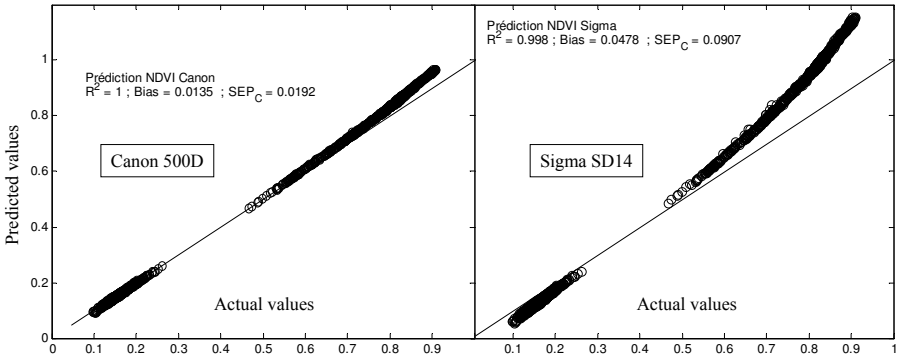


Fig. 5. Predicted versus reference values of NDVI

First, we can notice two clearly separated groups of NDVI values. They correspond respectively to soil and vegetation (wheat, monocotyledons and dicotyledons), confirming the remarkable efficiency of NDVI for crop-soil discrimination. The prediction quality using the Canon camera is significantly better: the NDVI values obtained are nearly equal to the reference ones, excepted for the highest values (NDVI > 0.8) where the error remains less than 10%.

5 Conclusion

We have shown in this paper that both NIR and R bands for NDVI computation can be obtained simultaneously from a single standard digital RGB still camera, by replacing the near-infrared blocking filter inside by a low-pass filter. A generic method, called BSOP, has been proposed to determine the optimal replacing filter. This method has been applied using real spectrometric data on

two commercial SLR cameras based respectively on a Bayer matrix sensor and a Foveon sensor, associated with a standard Wratten filter. The simulated NDVI values obtained for a large number of luminance spectra of vegetation and soil have been compared with reference NDVI values. The results obtained with the Bayer matrix sensor are particularly satisfactory. The Foveon provides a much less accurate NDVI prediction, but remains a pertinent choice in the context of vegetation/soil discrimination by NDVI thresholding. The results open new possibilities in terms of high-resolution imaging for crop monitoring. Our next step will be the real implementation and test of both types of devices on an UAV, in the frame of a weed monitoring application.

Acknowledgements. The research leading to these results has received funding from the European Unions Seventh Framework Programme [FP7/2007-2013] under grant agreement *n*^o 245986. We acknowledge *Lavion jaune* (<http://www.lavionjaune.fr>) for having provided the modified cameras and contributed in their spectrometric characterization.

References

1. Rouse, J.W., et al.: Monitoring vegetation systems in the great plains with ERTS. In: Third ERTS Symposium (1973)
2. Huete, A.R., et al.: A comparison of vegetation indices over a global set of TM images for EOS-MODIS. *Remote sensing of environment* 59(3), 440–451 (1997)
3. Jindong, W., Dong, W., Bauer, M.E.: Assessing broadband vegetation indices and QuickBird data in estimating leaf area index of corn and potato canopies. *Field Crops Research* 102(1), 33–42 (2007)
4. Zhengwei, Y., et al.: A Comparison of Vegetation Indices for Corn and Soybean Vegetation Condition Monitoring. In: *Geoscience and Remote Sensing Symposium, IGARSS 2009, Cape Town* (2009)
5. Lebourgeois, V., et al.: Can Commercial Digital Cameras Be Used as Multispectral Sensors? A Crop Monitoring Test. *Sensors* 8(11), 7300–7322 (2008)
6. Dare, P.M.: Small format digital sensors for aerial imaging applications. In: *XXIst ISPRS Congress, Beijing, China* (2008)
7. Yuhas, R.H., Goetz, A.F.H., Boardman, J.W.: Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm. In: *Summaries of 3rd Annual JPL Airborne Geoscience Workshop, vol. 1*, pp. 147–149. JPL Publication 92-14 (1992)
8. Vigneau, N., et al.: Potential of field hyperspectral imaging as a non destructive method to assess leaf nitrogen content in Wheat. *Field Crops Research* 122(1), 25–31 (2011)

Combining Neighbourhoods in Fuzzy Job Shop Problems

Jorge Puente¹, Camino R. Vela¹, and Inés González-Rodríguez²

¹ A.I. Centre and Department of Computer Science,
University of Oviedo, Spain
{puente, crvela}@uniovi.es
<http://www.aic.uniovi.es/tc>

² Department of Mathematics, Statistics and Computing,
University of Cantabria, Spain
ines.gonzalez@unican.es

Abstract. In the sequel, we propose a new neighbourhood structure for local search for the fuzzy job shop scheduling problem, which is a variant of the well-known job shop problem, where uncertain durations are modelled as fuzzy numbers and the objective is to minimise the expected makespan of the resulting schedule. The new neighbourhood structure is based on changing the position of a task in a critical block. We provide feasibility conditions and a makespan estimate which allows to select only feasible and promising neighbours. The experimental results illustrate the success of our proposal in reducing expected makespan within a memetic algorithm. The experiments also show that combining the new structure with an existing neighbourhood from the literature considering both neighborhoods at the same time, provides the best results.

1 Introduction

Scheduling forms an important body of research since the late fifties, with multiple applications in industry, finance and science [16]. Traditionally, it has been treated as a deterministic problem that assumes precise knowledge of all data. However, modelling real-world problems often involves processing uncertainty, for instance in activity durations. In the literature we find different proposals for dealing with ill-known durations [11]. Perhaps the best-known approach is to treat them as stochastic variables. An alternative is to use fuzzy numbers or, more generally, fuzzy intervals in the setting of possibility theory, which is said to provide a natural framework, simpler and less data-demanding than probability theory, for handling incomplete knowledge about scheduling data (c.f. [4]).

The complexity of scheduling problems such as job shop means that practical approaches to solving them usually involve heuristic strategies [2]. Extending these strategies to problems with fuzzy durations in general requires a significant reformulation of both the problem and solving methods. Proposals from the literature include a neural approach [20], genetic algorithms [18], [15], [7], simulated annealing [5] and genetic algorithms hybridised with local search [6], [9].

In this paper, we intend to advance in the study of local search methods to solve the job shop problem with task durations given as triangular fuzzy numbers and where the goal is to minimise the expected makespan, denoted $FuzJ||E[C_{max}]$. In [17] a neighbourhood \mathcal{N}_3 is proposed that, embedded in a memetic algorithm, notably reduces the computational load of local search with respect to a previous neighbourhood while maintaining or even improving solution quality. We shall propose a new neighbourhood structure, based on a definition of criticality from [9]. This will allow to obtain better quality solutions at the cost of increasing the number of neighbours. Even better, when it is used in conjunction with \mathcal{N}_3 it reaches even better solutions with a smaller set of neighbours and hence with a lower computational load. Finally, we propose that the local search be also integrated into the genetic algorithm framework.

2 Job Shop Scheduling with Uncertain Durations

The *job shop scheduling problem*, also denoted *JSP*, consists in scheduling a set of jobs $\{J_1, \dots, J_n\}$ on a set of physical resources or machines $\{M_1, \dots, M_m\}$, subject to a set of constraints. There are *precedence constraints*, so each job J_i , $i = 1, \dots, n$, consists of m tasks $\{\theta_{i1}, \dots, \theta_{im}\}$ to be sequentially scheduled. Also, there are *capacity constraints*, whereby each task θ_{ij} requires the uninterrupted and exclusive use of one of the machines for its whole processing time. A feasible schedule is an allocation of starting times for each task such that all constraints hold. The objective is to find a schedule which is *optimal* according to some criterion, most commonly that the *makespan* is minimal.

2.1 Uncertain Durations

In real-life applications, it is often the case that the exact time it takes to process a task is not known in advance, and only some uncertain knowledge is available. Such knowledge can be modelled using a *triangular fuzzy number* or TFN, given by an interval $[n^1, n^3]$ of possible values and a modal value n^2 in it. For a TFN N , denoted $N = (n^1, n^2, n^3)$, the membership function takes the following triangular shape:

$$\mu_N(x) = \begin{cases} \frac{x-n^1}{n^2-n^1} & : n^1 \leq x \leq n^2 \\ \frac{x-n^3}{n^2-n^3} & : n^2 < x \leq n^3 \\ 0 & : x < n^1 \text{ or } n^3 < x \end{cases} \quad (1)$$

In the job shop, we essentially need two operations on fuzzy numbers, the sum and the maximum. These are obtained by extending the corresponding operations on real numbers using the *Extension Principle*. However, computing the resulting expression is cumbersome, if not intractable. For the sake of simplicity and tractability of numerical calculations, we follow [5] and approximate the results of these operations, evaluating the operation only on the three defining points of each TFN. It turns out that for any pair of TFNs M

and N , the approximated sum $M + N \approx (m^1 + n^1, m^2 + n^2, m^3 + n^3)$ coincides with the actual sum of TFNs; this may not be the case for the maximum $\max(M, N) \approx (\max(m^1, n^1), \max(m^2, n^2), \max(m^3, n^3))$, although they have identical support and modal value.

The membership function of a fuzzy number can be interpreted as a possibility distribution on the real numbers. This allows to define its expected value [12], given for a TFN N by $E[N] = \frac{1}{4}(n^1 + 2n^2 + n^3)$. It coincides with the neutral scalar substitute of a fuzzy interval and the centre of gravity of its mean value [4]. It induces a total ordering \leq_E in the set of fuzzy numbers [5], where for any two fuzzy numbers M, N $M \leq_E N$ if and only if $E[M] \leq E[N]$.

2.2 Fuzzy Job Shop Scheduling

A job shop problem instance may be represented by a directed graph $G = (V, A \cup D)$. V contains one node $x = m(i - 1) + j$ per task θ_{ij} , $1 \leq i \leq n$, $1 \leq j \leq m$, plus two additional nodes 0 (or *start*) and $nm + 1$ (or *end*), representing dummy tasks with null processing times. Arcs in A , called *conjunctive arcs*, represent precedence constraints (including arcs from node *start* to the first task of each job and arcs from the last task of each job to node *end*). Arcs in D , called *disjunctive arcs*, represent capacity constraints; $D = \cup_{j=1}^m D_j$, where D_i corresponds to machine M_i and includes two arcs (x, y) and (y, x) for each pair x, y of tasks requiring that machine. Each arc (x, y) is weighted with the processing time p_x of the task at the source node (a TFN in our case). A feasible task processing order σ is represented by a *solution graph*, an acyclic subgraph of G , $G(\sigma) = (V, A \cup R(\sigma))$, where $R(\sigma) = \cup_{i=1 \dots m} R_i(\sigma)$, $R_i(\sigma)$ being a hamiltonian selection of D_i . Using forward propagation in $G(\sigma)$, it is possible to obtain the starting and completion times for all tasks and, therefore, the schedule and the makespan $C_{max}(\sigma)$.

The schedule will be fuzzy in the sense that the starting and completion times of all tasks and the makespan are TFNs, interpreted as possibility distributions on the values that the times may take. However, the task processing ordering σ that determines the schedule is crisp; there is no uncertainty regarding the order in which tasks are to be processed.

Given that the makespan is a TFN and neither the maximum nor its approximation define a total ordering in the set of TFNs, it is necessary to reformulate what is understood by “minimising the makespan”. In a similar approach to stochastic scheduling, it is possible to use the concept of expected value for a fuzzy quantity and the total ordering it provides, so the *objective* is to minimise the expected makespan $E[C_{max}(\sigma)]$, a crisp objective function.

Another concept that needs some reformulation in the fuzzy case is that of criticality, an issue far from being trivial. In [5], an arc (x, y) in the solution graph is taken to be critical if and only if the completion time of x and the starting time of y coincide in any of their components. In [9], it is argued that this definition yields some counterintuitive examples and a more restrictive notion is proposed. From the solution graph $G(\sigma)$, three *parallel solution graphs* $G^i(\sigma)$, $i = 1, 2, 3$, are derived with identical structure to $G(\sigma)$, but where the cost of arc

$(x, y) \in A \cup R(\sigma)$ in $G^i(\sigma)$ is p_x^i , the i -th component of p_x . Each parallel solution graph $G^i(\sigma)$ is a disjunctive graph with crisp arc weights, so in each of them a critical path is the longest path from node *start* to node *end*. For the fuzzy solution graph $G(\sigma)$, a path will be considered to be *critical* if and only if it is critical in some $G^i(\sigma)$. Nodes and arcs in a critical path are termed critical and a critical path is naturally decomposed into critical blocks, these being maximal subsequences of tasks requiring the same machine.

In order to simplify expressions, we define the following notation for a feasible schedule. For a solution graph $G(\pi)$ and a task x , let $P\nu_x$ and $S\nu_x$ denote the predecessor and successor nodes of x on the machine sequence (in $R(\pi)$) and let PJ_x and SJ_x denote the predecessor and successor nodes of x on the job sequence (in A). The *head* of task x is the starting time of x , a TFN given by $r_x = \max\{r_{PJ_x} + p_{PJ_x}, r_{P\nu_x} + p_{P\nu_x}\}$, and the *tail* of task x is the time lag between the moment when x is finished until the completion time of all tasks, a TFN given by $q_x = \max\{q_{SJ_x} + p_{SJ_x}, q_{S\nu_x} + p_{S\nu_x}\}$.

3 Improved Local Search

Part of the interest of critical paths stems from the fact that they may be used to define neighbourhood structures for local search. Roughly speaking, a typical local search schema starts from a given solution, calculates its neighbourhood and then neighbours are evaluated in the search of an improving solution. In simple hill-climbing, the first improving neighbour found will replace the original solution, so local search starts again from that improving neighbour. The procedure finishes when no neighbour satisfies the acceptance criterion. Clearly, a central element in any local search procedure is the definition of neighbourhood.

Neighbourhood structures have been used in different metaheuristics to solve the fuzzy job shop. In [5], a neighbourhood is used in a simulated annealing algorithm. The same neighbourhood is used in [6] for a memetic algorithm (MA) hybridising a local search procedure (LS) with a genetic algorithm (GA) using permutations with repetition as chromosomes. Results in [6] show that the hybrid method compares favourably with the simulated annealing from [5] and a GA from [18]. The same memetic algorithm is used in [9], but here the local search procedure uses the neighbourhood based on parallel graphs. The experimental results reported in [9] show that this new memetic algorithm performs better than state-of-the-art algorithms. Despite satisfactory, the results also suggest that the algorithm has reached its full potential and, importantly, most of the computational time it requires corresponds to the local search. In [8] and [17] two new neighbourhood structures have been defined. Both reduce the computational cost of the local search, specially the one from [17], while keeping similar or even identical quality of solutions.

In the following, we propose to improve local search efficiency in two steps. A first idea is to introduce in the local search algorithm a new neighbourhood structure, based on inserting a critical task into other position of its critical block, evaluating neighbours in a efficient manner and using makespan estimators.

A second idea, previously used in the crisp framework in [13], is to use the new structure together with a previous one to obtain an advanced neighbourhood definition which combines both their advantages.

3.1 Previous Approaches

A well-known neighbourhood for the deterministic job shop is that proposed in [21]. Given a task processing order π , its neighbourhood structure is obtained by reversing all the critical arcs in $G(\pi)$. This structure was first extended to the fuzzy case in [5], where a disjunctive arc (x, y) was taken to be critical in $G(\pi)$ if exists $i = 1, 2, 3$ such that $r_x^i + p_x^i = q_y^i$, i.e, the completion time of x coincides with the starting time of y in one component; the resulting neighbourhood will be denoted \mathcal{N}_0 in the following.

A second extension to the fuzzy case was proposed in [9], using the definition of criticality based on parallel solution graphs instead. Let us denote the resulting neighbourhood by \mathcal{N}_1 . As a consequence of the criticality definitions, $\mathcal{N}_1 \subset \mathcal{N}_0$ and any neighbour $\sigma \in \mathcal{N}_0 - \mathcal{N}_1$ can never improve the expected makespan of the original solution. Additionally, all neighbours in \mathcal{N}_1 are feasible and the connectivity property holds: starting from any solution, it is possible to reach a given global optimum in a finite number of steps using this structure. The experimental results endorsed the good theoretical behaviour, obtaining better expected makespan values than previous approaches from the literature. However, the large size of the structure for the fuzzy case resulted in an extremely high computational load.

To improve on efficiency, a reduced structure, denoted \mathcal{N}_2 in the following, was proposed in [8], inspired in the proposal for the deterministic problem from [14]. The neighbourhood was based on reversing only those critical arcs at the extreme of critical blocks of a single path, so $\mathcal{N}_2 \subset \mathcal{N}_1$. Clearly, \mathcal{N}_2 contains only feasible neighbours, although connectivity fails to hold. It was proved that the reversal of a critical arc (x, y) can only lead to an improvement if (x, y) is at the extreme of a critical block, and therefore, all neighbours from $\mathcal{N}_1 - \mathcal{N}_2$ are non-improving solutions. The experimental results showed how \mathcal{N}_2 resulted in a much more efficient search obtaining the same expected makespan values as with \mathcal{N}_1 . However, due to the fact that arcs may be critical on three different components, the neighbourhood size is still quite large and there is still room for improvement. It is also interesting to define different structures which allow for searching in different areas of the solution space.

All these neighbourhood structures were based on reversing a single critical arc. In [17], a new neighbourhood structure obtained by “inverting more than one arc”, that is, permuting the relative ordering of more than two consecutive tasks within a critical block, was proposed. Given a task processing order π and a critical arc (x, y) in the associated graph $G(\pi)$, $\mathcal{N}_3(\pi)$ is obtained by considering all possible permutations of the sequences $(P\nu_x, x, y)$ and $(x, y, S\nu_y)$ where the relative order between x and y is reversed.

For the aforementioned structures it is clear that $\mathcal{N}_2 \subset \mathcal{N}_1 \subset \mathcal{N}_3$. Moreover, \mathcal{N}_3 verifies the connectivity property and, as \mathcal{N}_1 , contains many not-improving

neighbours. A reduced neighbourhood, \mathcal{N}_3^R , is defined using only the extreme of critical blocks. \mathcal{N}_3^R contains \mathcal{N}_2 while covering a greater portion of promising areas in the search space. In principle, \mathcal{N}_3^R may contain unfeasible neighbours, so a method lpath is provided in [17] that allows to obtains a lower bound of the expected makespan of feasible neighbours, which is later used in order to always select feasible neighbours. Despite its larger search domain, this new structure notably reduces the computational load of local search with respect to the previous neighbourhood while maintaining solution quality.

3.2 New Neighbourhood Definition

All the neighbourhood structures proposed up to date are based on reversing one or more critical arcs. In the following, we propose a new neighbourhood structure obtained by inserting a critical task in another position in its critical block, a proposal inspired in the work for deterministic job shop from [3].

For a task x inside a block $b = (b', x, b'')$, where b' and b'' are sequences of tasks, the aim of the new neighbourhood is to move x to the first or the last position in b . Actually such moves may lead to infeasible solutions; if this is the case, we consider the closest move to the first or the last position for which feasibility is preserved.

Testing the feasibility of a solution may be computationally expensive. The next proposition, inspired in [3], gives a sufficient condition for feasibility after moving an operation x in a critical block towards the beginning of such block.

Proposition 1. *Let σ be a feasible processing order and let $b = (b'_1 \ b'_2 \ x \ b'')$ be a critical block in $G^i(\sigma)$ for some i , where b'_1 , b'_2 and b'' are sequences of tasks, a sufficient condition for feasibility of a solution $\pi = \sigma_{(b'_1, x, b'_2, b'')}$ is that*

$$\exists j = 1, 2, 3, \quad r_{PJ_x}^j < r_{SJ_y}^j + p_{SJ_y}^j \quad \forall y \in b'_2 \tag{2}$$

The proof of this proposition follows from the fact of that feasibility is lost if a cycle in the resulting digraph exist, and this cycle can only exist if and only if there exists an alternative path from a task in b'_2 to PJ_x . This property suggests the following definition of neighbourhood.

Definition 1 ($\mathcal{N}_4(\pi)$). *Let π be a task processing order and let x an operation in a critical block b . In a neighboring solution x is moved closest to the first or the last operation of b for which the sufficient condition of feasibility given by proposition [1] is preserved.*

Theorem 1. \mathcal{N}_4 *verifies the connectivity property: given a globally optimal processing order π_0 , it is possible to build a finite sequence of transitions of \mathcal{N}_4 starting from any non-optimal task processing order π and leading to π_0 .*

The proof of this property is ommited due to space constraints.

Notice however that the considerations reported in [10] about the so called *elimination properties* for the deterministic job shop are applicable here, making it advisable that \mathcal{N}_4 be reduced. Indeed, the insertion of a critical task x inside a

block in other position can only lead to an improvement if the new position is at the extreme of the block. This motivates the definition of the following reduced neighbourhood:

Definition 2. *Let π be a task processing order and let x be a task in a critical block b in the associated graph $G(\pi)$. Then, in a neighboring solution of the reduced neighbourhood structure, $\mathcal{N}_4^R(\pi)$, x is moved to the first or the last operation of b whenever the sufficient condition of feasibility given by proposition 1 is preserved.*

3.3 Makespan Estimation

In a monotonic local search method, as hill climbing used in this work, only those neighbours with improving makespan are of interest. Hence a makespan estimation may help reduce the computational cost of local search by discarding uninteresting neighbours without actually evaluating them. For the case when only one arc (x, y) is reversed, $\sigma_1 = \pi_{(y,x)}$, a lower bound of the neighbour’s makespan may be obtained by computing the length of the longest path in $G(\sigma_1)$ containing either x or y [19]. This can be done quickly (in time $O(nm)$) using heads and tails. In [17] this idea has been extended to every neighbour σ in $\mathcal{N}_3^R(\pi)$, by computing the length of a longest path in $G(\sigma)$ containing at least one of the nodes involved in the move. This is still valid for \mathcal{N}_4^R if we consider the sequence of tasks $X = (x_1, \dots, x_s)$ whose relative order has been permuted, although the method provides an estimate which is not necessarily a lower bound.

4 Experimental Results

We now consider 12 benchmark problems for job shop: the well-known FT10 and FT20, and the set of 10 problems identified in [1] as hard to solve for classical JSP: La21, La24, La25, La27, La29, La38, La40, ABZ7, ABZ8, and ABZ9. Ten fuzzy versions of each benchmark are generated following [5] and [9], so task durations become symmetric TFNs where the modal value is the original duration, ensuring that the optimal solution to the crisp problem provides a lower bound for the fuzzified version. In total, we consider 120 fuzzy job shop instances, 10 for each of the 12 crisp benchmark problems.

The goal of this section is to evaluate empirically our proposals. We consider the memetic algorithm (MA) presented in [17] which improved previous approaches from the literature in terms of makespan optimisation and efficiency. This algorithm combines a genetic algorithm with a simple hill-climbing local search procedure based on the neighbourhood structure \mathcal{N}_3^R . We shall use it as a baseline algorithm and introduce the different structures in the local search module: \mathcal{N}_3^R , \mathcal{N}_4^R and also, following the work from [13] for deterministic JSP, $\mathcal{N}_3^R \cup \mathcal{N}_4^R$. We have run the MA using the same parameters as in [17] (population size 100 and 200 generations). Table 1 shows for each MA version the average

Table 1. Results of MA using \mathcal{N}_3^R , \mathcal{N}_4^R , and $\mathcal{N}_3^R \cup \mathcal{N}_4^R$. CPU times are seconds (C++, Xeon E5520 2.26GHz).

Problem	Size	MA(\mathcal{N})	$RE_E[C_{max}]$			%Neigh.Inc	CPU
			Best	Avg	Worst		
FT10	10×10	\mathcal{N}_3^R	0.41	0.80	2.26		2.88
		\mathcal{N}_4^R	0.41	0.72	1.74	64.4	4.39
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	0.41	0.70	1.78	48.5	4.19
FT20	20×5	\mathcal{N}_3^R	0.03	0.70	1.13		3.80
		\mathcal{N}_4^R	0.03	0.31	1.12	166.6	8.14
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	0.03	0.43	1.12	104.6	6.81
La21	15×10	\mathcal{N}_3^R	0.88	1.16	1.37		5.05
		\mathcal{N}_4^R	0.85	1.07	1.29	64.8	7.66
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	0.77	1.06	1.27	41.1	6.99
La24	15×10	\mathcal{N}_3^R	0.71	1.24	2.07		4.93
		\mathcal{N}_4^R	0.63	1.11	1.49	60.6	7.24
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	0.69	1.15	1.50	39.3	6.77
La25	15×10	\mathcal{N}_3^R	0.28	0.77	1.19		5.01
		\mathcal{N}_4^R	0.27	0.82	1.11	89.7	8.03
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	0.26	0.78	1.10	53.2	7.40
La27	20×10	\mathcal{N}_3^R	0.89	2.14	2.75		8.94
		\mathcal{N}_4^R	0.68	1.77	2.52	107.9	15.69
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	0.62	1.71	2.47	61.8	13.79
La29	20×10	\mathcal{N}_3^R	1.87	3.47	4.90		8.48
		\mathcal{N}_4^R	1.39	2.81	4.06	108.9	14.62
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	1.41	2.71	4.21	63.1	12.57
La38	15×15	\mathcal{N}_3^R	1.06	2.12	4.16		8.86
		\mathcal{N}_4^R	0.98	2.31	3.96	63.2	13.25
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	0.95	2.24	4.01	33.3	11.87
La40	15×15	\mathcal{N}_3^R	0.82	1.36	2.03		9.17
		\mathcal{N}_4^R	0.92	1.38	2.12	67.2	13.93
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	0.86	1.32	1.96	34.1	12.64
ABZ7	20×15	\mathcal{N}_3^R	2.69	3.93	4.95		15.66
		\mathcal{N}_4^R	2.47	3.52	4.54	130.7	26.29
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	2.36	3.48	4.52	67.3	22.26
ABZ8	20×15	\mathcal{N}_3^R	6.22	7.58	8.89		16.97
		\mathcal{N}_4^R	5.81	7.15	8.52	141.1	31.60
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	5.65	7.01	8.35	68.0	25.22
ABZ9	20×15	\mathcal{N}_3^R	5.54	7.23	8.95		15.92
		\mathcal{N}_4^R	4.84	6.61	8.26	103.3	26.62
		$\mathcal{N}_3^R \cup \mathcal{N}_4^R$	4.67	6.51	8.15	51.7	22.56

across each family of ten fuzzy instances of the makespan relative error and of the variation neighbourhood size and the CPU time taken in average by one run. The relative error is calculated w.r.t. the value of the optimal solution of the crisp instance or to a lower bound when the optimal solution is not known.

The results show that \mathcal{N}_4^R breaks the existing quality threshold from [17], improving the relative error of the expected makespan in the best, average and worst solution for almost every instance. In average across all instances the improvement are 7.48%, 10.65% and 9.24% respectively. As expected, the tradeoff is the increase in the number of evaluated neighbours (97.37%) and hence in the CPU time required (66.37%).

Similarly to [13] for the deterministic JSP, both neighbourhoods are combined into an advanced one $\mathcal{N}_3^R \cup \mathcal{N}_4^R$, combining the advantages of both. Such combination may be expected to contain more neighbours and hence require more CPU time. However, it reaches better solutions than \mathcal{N}_4^R evaluating less neighbours. The increase in the number of neighbours evaluated by the MA compared to using \mathcal{N}_3^R is approximately 55%. Additionally, a *t*-test has been run to compare neighbourhood choices, namely, \mathcal{N}_3^R vs. \mathcal{N}_4^R , \mathcal{N}_3^R vs. $\mathcal{N}_3^R \cup \mathcal{N}_4^R$ and \mathcal{N}_4^R vs. $\mathcal{N}_3^R \cup \mathcal{N}_4^R$, using in all cases average makespan values. The results show the existence of statistically significant differences for each choice, with *p*-value=0.01 in all cases.

There is no clear correlation between instance sizes and makespan results. Instances with 20 jobs, with a large reduction in relative error also have an important increase in the number of neighbours. For square instances of size 15×15 , the MA with \mathcal{N}_3^R is better in average than with \mathcal{N}_4^R and sometimes also better than with $\mathcal{N}_3^R \cup \mathcal{N}_4^R$, but this is not the case in all square instances, as we can see for the FT10.

5 Conclusions

We have considered a job shop problem with uncertain durations modelled as TFNs. We have proposed a new neighbourhood structure for local search, denoted \mathcal{N}_4 , based on inserting a critical task into the most extreme position within its block which maintains feasibility. To do this, a sufficient condition for feasibility is provided and the resulting neighbourhood is shown to asymptotically converge to an optimum. A reduced neighbourhood, \mathcal{N}_4^R is obtained by allowing insertion only if it is at the extreme of the block. This allows to reduce the set of neighbours by pruning non-improving moves. Finally, experimental results show the good behaviour of this neighbourhood within a memetic algorithm. The experiments also show that combining \mathcal{N}_4^R with an existing neighbourhood structure from the literature we improve the best results so far whilst considerably reducing neighbourhood size and hence, the CPU time required.

Acknowledgements. All authors are supported by MEC-FEDER Grant TIN2010-20976-C02-02 .

References

1. Applegate, D., Cook, W.: A computational study of the job-shop scheduling problem. *ORSA Journal of Computing* 3, 149–156 (1991)
2. Brucker, P., Knust, S.: *Complex Scheduling*. Springer, Heidelberg (2006)

3. Dell' Amico, M., Trubian, M.: Applying tabu search to the job-shop scheduling problem. *Annals of Operational Research* 41, 231–252 (1993)
4. Dubois, D., Fargier, H., Fortemps, P.: Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. *European Journal of Operational Research* 147, 231–252 (2003)
5. Fortemps, P.: Jobshop scheduling with imprecise durations: a fuzzy approach. *IEEE Transactions of Fuzzy Systems* 7, 557–569 (1997)
6. González Rodríguez, I., Vela, C.R., Puente, J.: Sensitivity Analysis for the Job Shop Problem with Uncertain Durations and Flexible Due Dates. In: Mira, J., Álvarez, J.R. (eds.) *IWINAC 2007*. LNCS, vol. 4527, pp. 538–547. Springer, Heidelberg (2007)
7. González Rodríguez, I., Puente, J., Vela, C.R., Varela, R.: Semantics of schedules for the fuzzy job shop problem. *IEEE Transactions on Systems, Man and Cybernetics, Part A* 38(3), 655–666 (2008)
8. González Rodríguez, I., Vela, C.R., Hernández-Arauzo, A., Puente, J.: Improved local search for job shop scheduling with uncertain durations. In: *Proc. of ICAPS 2009*, pp. 154–161. AAAI Press (2009)
9. González Rodríguez, I., Vela, C.R., Puente, J., Varela, R.: A new local search for the job shop problem with uncertain durations. In: *Proc. of ICAPS 2008*, pp. 124–131. AAAI Press (2008)
10. Grabowski, J., Wodecki, M.: A very fast tabu search algorithm for job shop problem. In: *Metaheuristic Optimization via Memory and Evolution. Tabu Search and Scatter Search*. Operations Research/Computer Science Interfaces Series, pp. 117–144. Springer, Heidelberg (2005)
11. Herroelen, W., Leus, R.: Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research* 165, 289–306 (2005)
12. Liu, B., Liu, Y.K.: Expected value of fuzzy variable and fuzzy expected value models. *IEEE Transactions on Fuzzy Systems* 10, 445–450 (2002)
13. Mattfeld, D.C.: *Evolutionary Search and the Job Shop Investigations on Genetic Algorithms for Production Scheduling*. Springer, Heidelberg (1995)
14. Nowicki, E., Smutnicki, C.: A fast taboo search algorithm for the job shop scheduling problem. *Management Science* 42, 797–813 (1996)
15. Petrovic, S., Fayad, S., Petrovic, D.: Sensitivity analysis of a fuzzy multiobjective scheduling problem. *International Journal of Production Research* 46(12), 3327–3344 (2007)
16. Pinedo, M.L.: *Scheduling. Theory, Algorithms, and Systems*, 3rd edn. Springer, Heidelberg (2008)
17. Puente, J., Vela, C.R., González-Rodríguez, I.: Fast local search for fuzzy job shop scheduling. In: *Proc. of ECAI 2010*, pp. 739–744. IOS Press (2010)
18. Sakawa, M., Kubota, R.: Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms. *European Journal of Operational Research* 120, 393–407 (2000)
19. Taillard, E.D.: Parallel taboo search techniques for the job shop scheduling problem. *ORSA Journal on Computing* 6(2), 108–117 (1994)
20. Tavakkoli-Moghaddam, R., Safei, N., Kah, M.: Accessing feasible space in a generalized job shop scheduling problem with the fuzzy processing times: a fuzzy-neural approach. *Journal of the Operational Research Society* 59, 431–442 (2008)
21. Van Laarhoven, P., Aarts, E., Lenstra, K.: Job shop scheduling by simulated annealing. *Operations Research* 40, 113–125 (1992)

Learning Cooperative TSK-0 Fuzzy Rules Using Fast Local Search Algorithms

Javier Cózar, Luis de la Ossa, and Jose M. Puerta

Computing Systems Department, I^3A
University of Castilla-La Mancha, Spain
{javier.cozar, luis.delaozza, jose.puerta}@uclm.es

Abstract. This paper presents an adaptation of the COR methodology to derive the rule base in TSK-type linguistic fuzzy rule-based systems. In particular, the work adapts an existing local search algorithm for Mamdani rules which was shown to find the best solutions, whilst reducing the number of evaluations in the learning process.

Keywords: Fuzzy modeling, evolutionary fuzzy systems.

1 Introduction

There are many algorithms which learn the rules of FRBSs from data. Some of them, the initial approaches, evaluate each candidate rule in an independent manner [11] and, according to that evaluation, select the final set of rules. Despite the fact that the information in a single rule is relevant, the output produced by a FRBS when processing an input is generally obtained as a combination of the outputs produced by each one of the fired rules. This fact is called *cooperation*, and is the key to the power and expressiveness of this kind of systems.

It is important for FRBS learning algorithms to consider cooperation among rules in order to obtain accurate systems. There are some proposals, such as the COR methodology [2], which do that by using a search algorithm to derive the whole rule base. The main problem of such a technique is that it must find complete rule sets, the number of which increases dramatically with the size of the problem and the number of labels used for the fuzzy variables. Moreover, the evaluation of each one of these sets of rules is very costly, since it implies building the rule base and processing the whole dataset.

The method described in [6] proposed the use of a local search algorithm to find the best combination of rules. The reasons for this were both the inherent locality properties existing in the problem of selecting fuzzy rules, and the low computational cost required to evaluate local changes. The results obtained show that the algorithm could find systems which improve on those found by a Genetic Algorithm, whereas the cost of the search was slightly lower.

In this work, we propose the adaptation of the said algorithm to the learning of TSK-0 fuzzy rules. In these rules, the consequent is a real number instead of a fuzzy set. Therefore, the concept of neighbourhood must be adapted. In contrast

with other methods such as least squares [9], this method allows the number of rules to be varied.

This paper is divided into 5 sections besides this introduction. Section 2 describes the TSK-0 FRBSs. Afterwards, the COR methodology and its adaptation to TSK-0 rules is explained in Section 3, and the local search algorithm is presented in Section 4. Section 5 describes the experimental study and the results obtained. Finally, in Section 6, some concluding remarks will be made.

2 TSK-0 Fuzzy Rule-Based Systems

Fuzzy rules in FRBSs are composed of predicates of the form X is F , where X is a domain variable and F is a fuzzy set defined over the domain of such a variable. In the case of Linguistic Fuzzy Rule-Based Systems (LFRBSs) [12][8], rules are composed of predicates of the form X is A , where A is a linguistic label, i. e., a term associated with a certain fuzzy set that has previously been defined in the domain of X , which is then called a linguistic variable. In contrast with Mamdani-type FRBSs with scatter partitions [5], where each fuzzy set F is defined in the rule itself, in LFRBSs a *DataBase* is used which contains the linguistic variables and the specification of the linguistic terms.

The use of linguistic variables makes LFRB models easier to interpret by human experts. However, the limitation in the number of fuzzy sets that can be used to build the rules (only those defined in the *DataBase*) limits their performance.

There are some variations of the original LFRBSs which aim to improve their accuracy with a small loss of interpretability. Takagi-Sugeno-Kang (TSK) rules [10] are the most important of these. In TSK systems, the consequent for each rule R_s is a polynomial function of the input variables $P_s(X_1, \dots, X_n)$. Therefore, a rule is specified as:

$$R_s : \text{If } X_1 \text{ is } A_1^s \& \dots \& X_n \text{ is } A_n^s \text{ then } Y = a_{s1}X_1 + \dots + a_{sn}X_n + b_s$$

The order of a TSK FRBS refers to the degree of P_s . Hence, in TSK-0 systems the consequent is a constant value and the rules are expressed as:

$$R_s : \text{If } X_1 \text{ is } A_1^s \& \dots \& X_n \text{ is } A_n^s \text{ then } Y = b_s$$

Given an example $e_l = (x_1^l, \dots, x_n^l, y^l)$, the output in a TSK system is obtained as the weighted average of the outputs produced by each individual rule R_s in the *RuleBase* \mathcal{RB}_o :

$$\hat{y}_o^l = \frac{\sum_{R_s \in \mathcal{RB}_o} h_s^l P_s(x_1^l, \dots, x_n^l)}{\sum_{R_s \in \mathcal{RB}_o} h_s^l}$$

where $h_s^l(e_l) = T(A_1^s(x_1^l), \dots, A_n^s(x_n^l))$ is the matching degree of the example e_l with R_s , and T is a T-norm. In TSK-0 rules, the expression can be reduced to

$$\hat{y}_o^l = \frac{\sum_{R_s \in \mathcal{RB}_o} h_s^l b_s}{\sum_{R_s \in \mathcal{RB}_o} h_s^l} \tag{1}$$

3 The Cooperative Rules Methodology for TSK-0 Rules

Ad Hoc Data-Driven methods which learn the *RuleBase* of a FRBS take two elements as input:

- A *DataSet* \mathcal{E} , such that $\mathcal{E} = \{e_1, \dots, e_l, \dots, e_N\}$, with $e_l = (x_1^l, \dots, x_n^l, y^l)$.
- A *Linguistic DataBase*, which contains the definition of the linguistic variables, their domains, the fuzzy partitions, and the fuzzy terms \mathcal{A}_i for each input variable X_i . In the case of Mamdani rules, the fuzzy terms \mathcal{B} for the output variable Y ¹ are also needed. However, this is not necessary for TSK-0 rules. In this case, consequents are real numbers $b_s \in [\min_Y, \max_Y]$, where \min_Y and \max_Y bound the domain of variable Y .

Taking both elements as a starting point, these methods basically generate a set of candidate linguistic rules from the labels in the database in such a way that every example $e_l \in \mathcal{E}$ matches at least one rule. Afterwards, or simultaneously with the described process, some of the candidate rules are chosen to compose the final rule base.

In a formal way, let $S_s = (A_1^s, \dots, A_n^s)$, with $A_i^s \in \mathcal{A}_i$, $s \in \{1, \dots, N_s\}$, be a subspace of the input domain. S_s contains each example $e_l \in \mathcal{E}$ such that $\mu_{A_1^s}(x_1^l) \cdot \dots \cdot \mu_{A_n^s}(x_n^l) \neq 0$. The number of possible input subspaces (and rules of the system) is $N_s = \prod_{i=1}^n |\mathcal{A}_i|$. However, *Ad Hoc Data-Driven* methods generate rules which *only* cover the set of positive input subspaces S^+ .

There are two different criteria which are used to determine S^+ . In the first case, which is denominated *Example-based*, each example e_l enables at most one input subspace. Therefore,

$$S_s \in S^+ \text{ if } \exists e_l \in \mathcal{E} | \forall i \in \{1, \dots, n\}, \forall A_i' \in \mathcal{A}_i, \mu_{A_i^s}(x_i^l) \geq \mu_{A_i'}(x_i^l)$$

i.e., a subspace S_s is considered to generate candidate rules if there is an example e_l such that the rule with the highest matching degree with e_l is R_s . This is the case of algorithms such as Wang and Mendel [11].

The second alternative generates S^+ from what is called a *FuzzyGrid*. In this case

$$S_s \in S^+ \text{ if } \exists e_l \in \mathcal{E} | \mu_{A_1^s}(x_1^l) \cdot \dots \cdot \mu_{A_n^s}(x_n^l) \neq 0,$$

so an input subspace S_s is considered to generate candidate rules if there is an example e_l in it. In *FuzzyGrid*-based methods each example e_l can generate more than one rule. This implies the generation of systems with a higher number of rules, but whose precision is usually higher.

In both cases, the goal is to generate the rule R_s which is defined from S_s by determining its consequent, and selecting a subset of them to form the final rule base \mathcal{RB} . In the case of TSK-0 rules, this task consists of setting the consequent b_s for each candidate rule R_s .

¹ In this study, we will only consider an output variable, although the number of these can be greater than 1.

Algorithms such as Wang and Mendel [11] or ALM [4], or the Inductive Method [9] in the case of TSK-0 modeling, choose the consequent for each rule in an independent manner. In the case of the Inductive Method, for each subspace $S_s \in S^+$ a rule R_s is generated such that

$$b_s = \frac{\sum_{e_l | h_s^l > 0} h_s^l y^l}{\sum_{e_l | h_s^l > 0} h_s^l}$$

Evaluating the individual performance of each rule, though, does not consider the way FRBSs process the inputs. As mentioned above, an input example $e_l = (x_1^l \dots x_n^l)$ fires each rule R_s such that $\mu_{A_1^s}(x_1^l) \dots \mu_{A_n^s}(x_n^l) > 0$. This produces a real value b_s as the output, and the matching degree h_s^l . Let \mathcal{R}^l be the set of rules R_s fired by the input e_l . The output \hat{y}^l is obtained through a combination of the outputs produced by each rule in \mathcal{R}^l (expression [1]). Thus, the real performance of a rule R_s can only be evaluated in the context of the set of rules which would be fired when processing each example e_l which fires R_s . Therefore, given R_s , if \mathcal{E}^s is the subset of examples e_l such that $\mu_{A_1^s}(x_1^l) \dots \mu_{A_n^s}(x_n^l) > 0$, the evaluation of R_s depends on those rules R_t such that $\mathcal{E}^s \cap \mathcal{E}^t \neq \emptyset$.

There are some approaches which consider such cooperation among rules. In the Cooperative Rules (COR) methodology [2] a subspace S_s can generate several candidate rules $R_s \in \{CR(S_s) \cup \emptyset\}$, and only one (or none) of them can be chosen to be a part of the rule base. Rules generated from S_s only differ in the consequents, which can be chosen from the set $C(S_s)$, which is also obtained from examples. Once the search space is defined, i.e., the candidate rules are generated, each solution \mathcal{RB}_o can be represented by a vector of discrete values $\mathbf{c}_o = \{c_1^o, \dots, c_s^o, \dots, c_{|S^+|}^o\}$, where $c_s^o \in \{0, \dots, |C(S_s)|\}$. If $c_s^o > 0$, this means that the rule built with the c_s^o th consequent in $C(S_s)$ is included in \mathcal{RB}_o . Otherwise, if $c_s^o = 0$, the rule will not be included in \mathcal{RB}_o .

As the search space can be codified by a finite set of discrete numbers, the search can be carried out by any combinatorial optimization algorithm, such as Genetic Algorithms [3] or Ant Colony Optimization [1].

Regardless of the algorithm used, each configuration or candidate rule base must be evaluated. In order to do this, any error measure can be used. In this study, we consider the Mean Squared Error (MSE). Let \hat{y}_o^l be the output produced by the LFRBS system which contains the set of rules \mathcal{RB}_o when processing the example e_l . The error produced by the system with the set of rules \mathcal{RB}_o when processing the data set \mathcal{E} is obtained as follows:

$$MSE_o(\mathcal{E}) = \sum_{l=1}^N \frac{(\hat{y}_o^l - y^l)^2}{N} \tag{2}$$

The adaptation of the COR method to the case of TSK-0 rules is straightforward. Instead of finding a consequent from a set of them, the goal in TSK-0 rules is to find the real consequents $b_s \in [min_Y, max_Y]$. Each solution \mathcal{RB}_o can then be represented by a vector of real values $\mathbf{b}_o = \{b_1^o, \dots, b_s^o, \dots, b_{|S^+|}^o\}$, and any search method which works over real domains can be used to find such consequents.

Lastly, it is important to point out that one of the values which can be taken by b_s^o must represent the non-inclusion of the rule R_s in the system.

4 Using Local Search to Find the Rule Base

The algorithm described in [6] uses local search algorithms to find the set of rules in the COR approach. This proposal is based on the fact that, given a rule base \mathcal{RB}_o , a change in one rule R_s^o may affect the optimal choice of the consequents for those rules R_t^o such that $\mathcal{E}^s \cap \mathcal{E}^t \neq \emptyset$. Although this change could potentially affect all rules in \mathcal{RB}_o , it rarely extends to others in practice. In that work, the neighbourhood of a rule base \mathcal{RB}_o is defined as

$$\mathcal{N}(\mathcal{RB}_o) = \{RB_p | \exists! S_s \text{ with } R_s^p \neq R_s^o, R_s^o, R_s^p \in CR(S_s), R_s^p \in \mathcal{RB}_p, R_s^o \in \mathcal{RB}_o\}$$

Therefore, the neighbours of a rule base \mathcal{RB}_o are obtained by changing the consequent of one of its rules R_s^o .

Figure 1 shows the scheme of the local search algorithm. Once an initial solution is generated, either randomly or by some other method as [11][4], the algorithm performs a *HillClimbing*, carrying out, at each step, the change which produces the highest decrement in the MSE. The algorithm stops when no change produces an improvement in the current configuration.

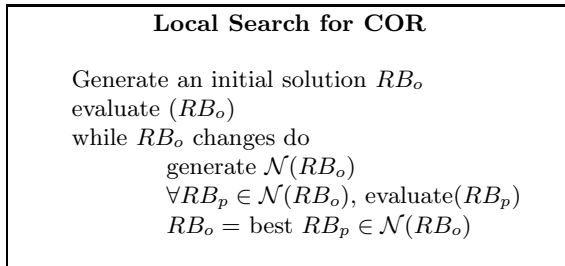


Fig. 1. Local Search algorithm for COR

Besides obtaining better results than a Genetic Algorithm in terms of accuracy, the algorithm is considerably more efficient. The evaluation of a configuration \mathbf{c}_o requires building the rule base \mathcal{RB}_o it codifies and then, for each example $e_l \in \mathcal{E}$, all rules $R_s^o \in \mathcal{RB}_o$ must be processed, producing the set of fuzzy outputs \mathcal{B}^l , which must be aggregated and defuzzified. However, in the case of local search, only one rule needs to be updated when calculating the error for each neighbour, and only the outputs for the examples covered by that rule need to be re-calculated to obtain the new MSE.

Furthermore, many other calculations can be avoided. Let $\mathcal{N}_s(\mathcal{RB}_o)$ be the subset of neighbours of \mathcal{RB}_o obtained by replacing R_s^o , and let $\mathcal{RB}_p \in \mathcal{N}_s(\mathcal{RB}_o)$. It is not necessary to evaluate all elements in $\mathcal{N}(\mathcal{RB}_p)$. In fact, if $\mathcal{E}^s \cap \mathcal{E}^t = \emptyset$, there is no need to calculate $\mathcal{N}_t(\mathcal{RB}_p)$. If $\mathcal{RB}_{o'} \in \mathcal{N}_t(\mathcal{RB}_o)$, and $\mathcal{RB}_{p'} \in \mathcal{N}_t(\mathcal{RB}_p)$,

then the difference between the errors produced by $\mathcal{RB}_{p'}$ and \mathcal{RB}_p is the same as that between $\mathcal{RB}_{o'}$ and \mathcal{RB}_o , which was already calculated. As is shown by the experimental results, this fact allows a huge number of evaluations to be avoided.

Extensions of local search algorithms can also be efficiently implemented. Thus, in [6] an algorithm is also presented that is based on Iterated Local Search [7]. It is depicted in figure 2. In this algorithm, once the local search is not able to find a better set of rules, the current solution is perturbed. This perturbation can be done by changing the consequents of some rules at random. However, there is an alternative aimed at decreasing the number of rules, which consists of setting those rules to the value \emptyset so that they are not included in the rule base. Once a rule is chosen, it is finally removed unless this produces an increment of more than 5% in the error.

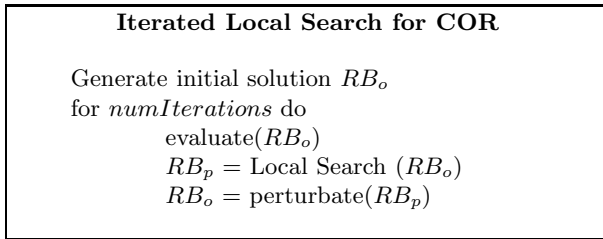


Fig. 2. Iterated Local Search algorithm for COR

4.1 Adaptation of COR Methodology to Learning TSK-0 Rules

As mentioned above, the neighbourhood of a rule system $\mathcal{N}(\mathcal{RB}_o)$ is obtained by changing the consequent of one rule. As the number of possible consequents is limited, so is the size of the neighbourhood, which is calculated as follows:

$$|\mathcal{N}(\mathcal{RB}_o)| = \sum_{s=1}^{|S^+|} |C(S_s)| - 1$$

This fact makes it possible to evaluate each possible neighbour in the case of Mamdani fuzzy systems. In the case of TSK-0 systems this is not possible, since each consequent is a real value and $|\mathcal{N}(\mathcal{RB}_o)| = \infty$. However, given a rule base \mathcal{RB}_o , it is possible to obtain the consequent of a rule $R_s^o \in \mathcal{RB}_o$ which produces the best reduction in MSE.

Let R_s^o be the rule whose neighbourhood must be calculated, $e_l \in \mathcal{E}^s$ all examples covered by that rule, and $\mathcal{R}^l \subset \mathcal{RB}_o$ the set of rules fired by e_l . The output produced by the system \mathcal{RB}_o when processing e_l is obtained as:

$$\hat{y}_o^l = \frac{\sum_{R_s^o \in \mathcal{R}^l} h_s^l b_s}{\sum_{R_s^o \in \mathcal{R}^l} h_s^l} = \frac{\sum_{R_t^o \in \mathcal{R}^l | R_t^o \neq R_s^o} h_t^l p_t + h_s^l b_s}{\sum_{R_t^o \in \mathcal{R}^l} h_t^l}$$

In order to simplify, it will be considered that $HB_{o-s}^l = \sum_{R_t^o \in \mathcal{R}^l | R_t^o \neq R_s^o} h_t^l b_t$ and $H_o^l = \sum_{R_t^o \in \mathcal{R}^l} h_t^l$. Therefore, the former equation is expressed as:

$$\hat{y}_o^l = \frac{HB_{o-s}^l + h_s^l b_s}{H_o^l}$$

The squared error produced by the system \mathcal{RB}_o when calculating the output for the example e_l is:

$$SE_o(e_l) = \left(\frac{HB_{o-s}^l + h_s^l b_s}{H_o^l} - y^l \right)^2$$

And the MSE produced by the system \mathcal{RB}_o when calculating the outputs for all the rules covered by the rule R_s is:

$$MSE_o(\mathcal{E}^s) = \sum_{e_l \in \mathcal{E}^s} \left(\frac{HB_{o-s}^l + h_s^l b_s}{H_o^l} - y^l \right)^2$$

As this expression is quadratic in b_s , only the minimum can be calculated. Since the minimum of this function is the point where $MSE_o(\mathcal{E}^s)' = 0$, the optimum consequent for the rule b_{s_o} can be obtained as:

$$b_{s_o} = \frac{\sum_{e_l \in \mathcal{E}^s} \left(\frac{2y_l H_o^l h_s^l}{H_o^{l^2}} \right) - \sum_{e_l \in \mathcal{E}^s} \left(\frac{2HB_{o-s}^l h_s^l}{H_o^{l^2}} \right)}{\sum_{e_l \in \mathcal{E}^s} \left(\frac{2h_s^{l^2}}{H_o^{l^2}} \right)} \tag{3}$$

It is worth pointing out that the values for each h_s^l need only be calculated once, since they do not change during the search process.

Lastly, there are only two possible values which are considered in order to calculate $\mathcal{N}_s(\mathcal{RB}_o)$: b_{s_o} and \emptyset . Therefore, the neighbourhood is small. Nevertheless, the algorithm may perform an extremely large number of iterations, although there is a point after which the improvements made are insignificant. There are two strategies that can solve this problem. The first one consists of stopping the search when the improvement produced does not reach a threshold. The second one, used in this work, avoids evaluating neighbours when the difference between the best consequent for a certain rule b_{s_o} and the consequent used in the current system does not reach a minimum threshold.

5 Experimental Study

In order to test the proposed algorithms, they have been used to model four datasets taken from the FMLib repository². Two of them, f_1 and f_2 , are synthetic functions with two input variables, whereas the other two are two real problems. The first one, *ele1*, consists of finding a model that relates the *total length of a low-voltage line* installed in a rural town to the *number of inhabitants in the town*

² <http://decsai.ugr.es/~casillas/fmlib/index.html>

and the *mean of the distances from the center of the town to the three furthest clients in it*. The goal is to use the model to estimate the total length of line being maintained. Therefore, there are two predictive variables and one output one. In relation to the other problem, *ele2*, the aim is to predict the minimum maintenance costs of the low-voltage line installed in a rural town. There are four input variables: *sum of the lengths of all streets in the town*, *total area of the town*, *area that is occupied by buildings*, and *energy supplied to the town*.

The local search algorithm for learning TSK-0 rules (LS-TSK0) has been implemented by following the scheme proposed in Section 4, and includes all the optimizations commented in 6 to avoid evaluations. As a starting point, it uses the rule base obtained by means of the Inductive Method (IMethod).

With regards to the Iterated Local Search algorithm (ILS-TSK0), it is based on the local search algorithm above, and performs 20 iterations. After each one, 10% of the rules in the rule base are randomly removed.

In order to make a comparison, both the local search algorithm (fast version) described in 6 (LS-COR), and the Inductive Method, have also been tested. In all cases, only the *FuzzyGrid*-based approach has been considered to generate the sets of candidate rules. Moreover, the fuzzy variables have been modeled using triangular symmetrical fuzzy partitions by considering 7 labels for each linguistic variable.

The results have been obtained by performing 30 independent executions over the dataset. On each run, the dataset was randomized using the same seed for all algorithms, and 20% of the examples were left for testing. Then, the training error, the test error, the number of rules in the final system, the number of rule bases evaluated, and the number of processed instances were obtained. In the case of ILS-TSK0, the number of evaluations or processed instances corresponds to the one in which the best system was found. The results are shown in Table 5.

As expected, the results obtained by the LS-TSK0 algorithm improve significantly on those obtained by LS-COR in terms of accuracy. Moreover, the number of evaluated systems and processed instances is also improved. This difference exists regardless of the problem, although it is more significant in the case of the real datasets. Lastly, the LS-TSK0 algorithm finds systems with a higher number of rules. This difference is more significant in the case of *ele2*.

Obviously, the proposed algorithm improves on the results obtained with the Inductive Method, since it takes this algorithm as a starting point. However, we have considered that including those results would be of interest. In particular, the number of rules in the systems produced by the Inductive Method is the same as would be obtained by the algorithms described in 4. As can be noticed, the local search algorithms remove a significant number of rules when dealing with the real-world datasets. This does not happen with the synthetic functions because the distribution of input data over the search space is homogeneous in this case.

In order to compare the errors obtained by the ILS-TSK0 and TSK-0 algorithms, a paired *t-test* was performed for the two algorithms. The test shows that the results obtained do not present a significant difference (with $\alpha = 0.95$) in any case except *ele1*³. However, ILS-TSK0 achieves a considerable reduction in the number of rules for problem *ele2*, where the number of rules is more dramatic. Thus, despite the fact that the average number of candidate rules is 550.4, and the local search algorithm reduces the size of the rule base to 489.4 rules, the ILS algorithm removes about 200 more rules.

Results for problem f1

Search algorithm	Training err.	Test err.	#Rules	# Ev. Sytems	#Proc. Instances
LS (COR)	1.713	1.771	48.9	832.4	95337.8
I.Method	2.437	2.492	49.0	1.0	1681.0
LS(TSK-0)	0.804	0.820	49.0	632.7	91383.9
ILS(TSK-0)	0.804	0.820	49.0	639.5	93010.1

Results for problem f2

Search algorithm	Training err.	Test err.	#Rules	# Ev. Sytems	#Proc. Instances
LS (COR)	0.380	0.392	46.8	461.4	17796.2
I.Method	0.613	0.658	49.0	1.0	593.0
LS(TSK-0)	0.159	0.158	47.0	241.3	11444.8
ILS(TSK-0)	0.157	0.209	46.6	1160.2	60232.5

Results for problem ele1

Search algorithm	Training err.	Test err.	#Rules	# Ev. Sytems	#Proc. Instances
LS (COR)	567.673	694.345	28.1	624.3	29393.6
I.Method	597.434	665.652	33.4	1	396.0
LS(TSK-0)	530.775	673.804	27.8	340.8	16824.2
ILS(TSK-0)	529.694	684.380	25.0	1139.4	56237.5

Results for problem ele2

Search algorithm	Training err.	Test err.	#Rules	# Ev. Sytems	#Proc. Instances
LS (COR)	242.831	277.375	440.5	22223.2	833605.7
I.Method	271.594	286.124	550.4	1.0	845.0
LS(TSK-0)	134.854	156.065	489.4	10212.2	472773.3
ILS(TSK-0)	132.738	158.259	287.4	41736.3	1188659.9

6 Final Conclusions and Future Work

This work presents an efficient local-search algorithm for learning TSK-0 rules which is an adaptation of another defined for Mamdani rules. This adaptation involved modifying both codification and neighbourhood. The results obtained show a significant improvement on the original version. Moreover, it has been possible to implement an ILS algorithm which reduces the number of rules in the final system for problems with higher dimensionality.

In further works, we aim to adapt this algorithm to TSK-1 rules, where the consequent is a polynomial function. Moreover, we plan to use some other local heuristic, such as greedy construction, so that the search is more effective.

³ Although it might have been expected, there is no statistical difference in f2.

Acknowledgments. This study has been partially aided by the Consejería de Educación y Ciencia (JCCM), Spanish Government (MICINN) and FEDER funds under projects PCI08-0048-857 and TIN2010-20900-C04-03.

References

1. Casillas, J., Cerdón, O., Fernández de Viana, I., Herrera, F.: Learning cooperative linguistic fuzzy rules using the best-worst ant system algorithm. *International Journal of Intelligent Systems* 20, 433–452 (2005)
2. Casillas, J., Cerdón, O., Herrera, F.: Cor: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics* 32(4), 526–537 (2002)
3. Casillas, J., Cerdón, O., Herrera, F.: Different approaches to induce cooperation in fuzzy linguistic models under the COR methodology. In: *Technologies for constructing intelligent systems: Tasks*, pp. 321–334. Physica-Verlag GmbH, Heidelberg (2002)
4. Cerdón, O., Herrera, F.: A proposal for improving the accuracy of linguistic modeling. *IEEE Transactions on Fuzzy Systems* 8(3), 335–344 (2000)
5. Cerdón, O., Herrera, F., Hoffmann, F., Magdalena, L.: *Genetic fuzzy systems: Evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific (2001)
6. de la Ossa, L., Gámez, J.A., Puerta, J.M.: Learning cooperative fuzzy rules using fast local search algorithms. In: *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010)*, pp. 2134–2141 (2006)
7. Glover, F.W., Kochenberger, G.A.: *Handbook of Metaheuristics*. International Series in Operations Research & Management Science. Springer, Heidelberg (2003)
8. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies* 7, 1–13 (1975)
9. Nozaki, K., Ishibuchi, H., Tanaka, H.: A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems* 86, 251–270 (1997)
10. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications for modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics* 15(1), 116–132 (1985)
11. Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics* 22(6), 1414–1427 (1992)
12. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning. *Information Science* 8, 199–249 (1975)

Weighted Tardiness Minimization in Job Shops with Setup Times by Hybrid Genetic Algorithm

Miguel A. González, Camino R. Vela, and Ramiro Varela

Department of Computer Science,
University of Oviedo, (Spain) Campus de Viesques s/n, Gijón, 33271, Spain
<http://www.aic.uniovi.es/tc>

Abstract. In this paper we confront the weighted tardiness minimization in the job shop scheduling problem with sequence-dependent setup times. We start by extending an existing disjunctive graph model used for makespan minimization to represent the weighted tardiness problem. Using this representation, we adapt a local search neighborhood originally defined for makespan minimization. The proposed neighborhood structure is used in a genetic algorithm hybridized with a simple tabu search method. This algorithm is quite competitive with state-of-the-art methods in solving problem instances from several datasets of both classical JSP and JSP with setup times.

1 Introduction

In this paper we confront the Job Shop Scheduling Problem with Sequence-Dependent Setup Times (SDST-JSP) with weighted tardiness minimization. JSP has interested to researchers over the last decades, but in most cases the objective function is the makespan, even though other objective functions such as weighted tardiness or total flow time are sometimes more important in many real-life problems. Also, setup considerations are a relevant characteristic of many real scheduling problems that add to the difficulty of solving these problems with respect to their non-setup counterparts. Incorporating sequence-dependent setup times changes the nature of scheduling problems, so well-known results and techniques for the JSP are not directly applicable to the SDST-JSP. Some extensions have been done for makespan minimization in [16,13].

As far as we know, the best approach to weighted tardiness minimization in the SDST-JSP is the presented by Sun and Noble in [10]. They propose a shifting bottleneck algorithm and in their experimental study this algorithm is compared with very simple heuristics (some priority rules) across randomly generated instances that are not available for further comparison. However, weighted tardiness has been largely considered for the classic JSP, maybe the algorithms proposed in [3] and [7] being the most relevant approaches currently. In [3], Es-safi et al. propose a hybrid genetic algorithm that uses a local search based in reversing critical arcs and an algorithm that iterates between hill climbing and random generation of neighbors to escape from local optima. In [7] Mati et al. propose a local search method that uses estimators to evaluate the neighbors and

that is capable of minimizing any regular objective function, i.e. a non-decreasing function on the completion time of the operations.

We propose a hybrid algorithm that combines a genetic algorithm (GA) with tabu search (TS). The core of this algorithm is a variation of the neighborhood structure N_1^S introduced in [13] for the SDST-JSP with makespan minimization, which in its turn extends the structures proposed in [12] for the classical JSP. We define a disjunctive graph model for the SDST-JSP with weighted tardiness minimization to formalize this neighborhood. The proposed algorithm is termed $GA + TS$ in the following. We also define a method for estimating the weighted tardiness of the neighbors, and we will see that this estimation is less accurate and more time consuming than similar estimations for the makespan, due to the difference in the problem difficulty. We have conducted an experimental study across conventional benchmarks to compare $GA + TS$ with state-of-the-art algorithms in both classical JSP and SDST-JSP. The results show that the proposed algorithm outperforms the other methods.

The rest of the paper is organized as follows. In Section 2 we formulate the JSP and introduce the notation used across the paper. In section 3 we describe the main components of the genetic algorithm. The proposed neighborhood structure, the weighted tardiness estimation algorithm and the main components of the TS algorithm are described in Section 4. Section 5 reports results from the experimental study. Finally, in Section 6 we summarize the main conclusions and propose some ideas for future work.

2 Description of the Problem

In the job shop scheduling problem, a set of N jobs, $J = \{J_1, \dots, J_N\}$, are to be processed on a set of M machines (resources), $R = \{R_1, \dots, R_M\}$ while minimizing some function of completion times of the jobs, subject to constraints: (i) the sequence of machines for each job is prescribed, and (ii) each machine can process at most one job at a time. The processing of a job on a machine is called an operation, and its duration is a given constant. We denote by p_u the processing time of operation u . A time may be needed to adjust a machine between two consecutive operations, which is called a setup time, and which may or may not be sequence-dependent. We adopt the following notation for the setup times: S_{uv} is the setup time between consecutive operations $u, v \in R_j$, and S_{0u} is the setup time required before u if this operation is the first one scheduled on his machine. A job J_i may also have a due date d_i , that is a time before jobs should be completed, and a weight w_i , that is the relevance of the job. The objective here is to minimize the weighted cost of the jobs exceeding its due-dates, also known as the weighted tardiness. In the following, we denote by t_u the starting time of operation u , that needs to be determined.

The SDST-JSP has two binary constraints: precedence and capacity. Precedence constraints, defined by the sequential routings of the tasks within a job, translate into linear inequalities of the type: $t_u + p_u \leq t_v$, if v is the next operation to u in the job sequence. Capacity constraints that restrict the use of each

resource to only one task at a time translate into disjunctive constraints of the form: $t_u + p_u + S_{uv} \leq t_v \vee t_v + p_v + S_{vu} \leq t_u$, where u and v are operations requiring the same machine.

The objective is to obtain a feasible schedule that minimizes the weighted tardiness, defined as:

$$\sum_{i=1, \dots, N} w_i T_i$$

where T_i is the tardiness of the job i , defined as $T_i = \max\{C_i - d_i, 0\}$, where C_i is the completion time of job i . This problem is denoted by $J|s_{ij}|\sum w_i T_i$ in the literature.

2.1 The Disjunctive Graph Model Representation

The disjunctive graph is a common representation in scheduling, its exact definition depending on the particular problem. For the $J|s_{ij}|\sum w_i T_i$ problem, we propose that it be represented by a directed graph $G = (V, A \cup E \cup I_1 \cup I_2)$. Each node in set V represents a task of the problem, with the exception of the dummy nodes *start* and end_i $1 \leq i \leq N$, which represent fictitious operations that do not require any machine. Arcs in A are called *conjunctive arcs* and represent precedence constraints while arcs in E are called *disjunctive arcs* and represent capacity constraints. Set E is partitioned into subsets E_i , with $E = \cup_{j=1, \dots, M} E_j$, where E_j corresponds to resource R_j and includes two directed arcs (v, w) and (w, v) for each pair v, w of operations requiring that resource. Each arc (v, w) in A is weighted with the processing time of the operation at the source node, p_v , and each arc (v, w) of E is weighted with $p_v + S_{vw}$. Set I_1 includes arcs of the form $(start, v)$ for each operation v of the problem, weighted with S_{0v} . Set I_2 includes arcs $(\omega(i), end_i)$, $1 \leq i \leq N$, weighted with $p_{\omega(i)}$, where $\omega(i)$ is the last operation of job J_i .

A feasible schedule is represented by an acyclic subgraph G_s of G , $G_s = (V, A \cup H \cup J_1 \cup I_2)$, where $H = \cup_{j=1 \dots M} H_j$, H_j being a minimal subset of arcs of E_j defining a processing order for all operations requiring R_j and where J_1 consists of arcs $(start, v_j)$, $j = 1 \dots M$, v_j being the first operation of H_j . Finding a solution can thus be reduced to discovering compatible orderings H_j , or partial schedules, that translate into a solution graph G_s without cycles. Figure 1 shows a solution to a problem with 3 jobs and 3 machines; dotted arcs belong to H and J_1 , while continuous arcs belong to A .

To calculate the weighted tardiness of the schedule we have to compute the cost of a critical path in G_s to each node end_i $1 \leq i \leq N$, i.e., a directed path in G_s from node *start* to node end_i having maximum cost. Nodes and arcs in a critical path are also termed critical. A critical path may be represented as a sequence of the form $start, B_1, \dots, B_r, end_i$, $1 \leq i \leq N$, where each B_k , $1 \leq k \leq r$, is a critical block, a maximal subsequence of consecutive operations in the critical path requiring the same machine.

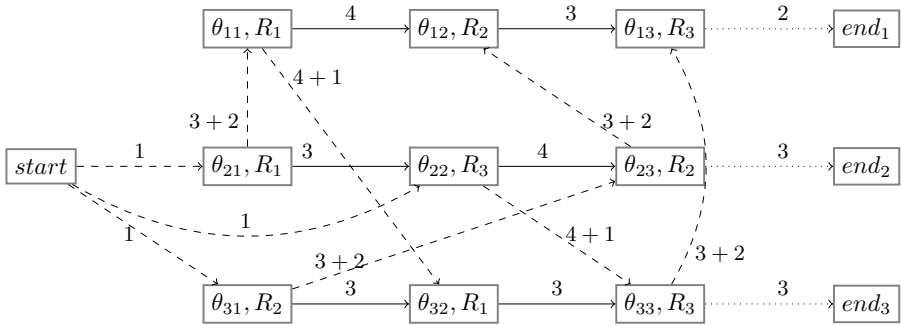


Fig. 1. A feasible schedule to a problem with 3 jobs and 3 machines

In order to simplify expressions, we define the following notation for a feasible schedule. Given a solution graph G_s for the SDST-JSP, the head of an operation v , denoted r_v , is the cost of the longest path from node $start$ to node v , i.e., the starting time of v in the schedule represented by G_s . A tail q_v^i , $1 \leq i \leq N$ is the cost of the longest path from node v to node end_i , minus the duration of task in node v . For practical reasons we will take $q_v^i = -\infty$ when no path exist from v to end_i . Here, it is important to remark that we have had to define N tails for each operation, while for makespan minimization it is required just one. Let PJ_v and SJ_v denote respectively the predecessor and successor of v in the job sequence, and PM_v and SM_v the predecessor and successor of v in its machine sequence. We take node $start$ to be PJ_v for the first task of every job and PM_v for the first task to be processed in each machine; note that $p_{start} = 0$. Then, the head of every operation v and every dummy node may be computed as follows:

$$\begin{aligned}
 r_{start} &= 0 \\
 r_v &= \max(r_{PJ_v} + p_{PJ_v}, r_{PM_v} + p_{PM_v} + S_{PM_v v}) \\
 r_{end_i} &= r_v + p_v, (v, end_i) \in I_2, 1 \leq i \leq N
 \end{aligned}$$

Similarly, for $1 \leq i \leq N$, we take node end_i as SJ_v for the last task of job i , and $p_{end_i} = 0$. So, the tail of all operations are computed as follows:

$$\begin{aligned}
 q_{end_i}^i &= 0 \\
 q_{end_i}^j &= -\infty, j \neq i \\
 q_v^j &= \begin{cases} \max(q_{SJ_v}^j + p_{SJ_v}, q_{SM_v}^j + p_{SM_v} + S_{vSM_v}) & \text{if } SM_v \text{ exists} \\ q_{SJ_v}^j + p_{SJ_v} & \text{otherwise} \end{cases} \\
 q_{start}^j &= \max_{v \in SM_{start}} \{q_v^j + p_v + S_{0v}\}
 \end{aligned}$$

3 Genetic Algorithm for the SDST-JSP

In this paper we will use a conventional GA, with permutations with repetition as our encoding schema and *Job Order Crossover* (JOX) for chromosome mating, uniform selection and generational replacement using tournament between two parents and two offspring. To build schedules we have used the Serial Schedule Generation Schema (SSGS) proposed in [1] for the SDST-JSP. SSGS iterates over the operations in the chromosome sequence and assigns each one the earliest starting time that satisfies all constraints with respect to previously scheduled operations. SSGS produces active schedules, provided that the triangular inequality for the setup times holds for all operations requiring the same machine [1], and this is the case in the instances used in our experimental study.

When combined with GA, TS is applied to every schedule produced by SSGS. Then, the chromosome is rebuilt from the improved schedule obtained by TS, so as its characteristics can be transferred to subsequent offspring. This effect of the evaluation function is known as Lamarckian evolution.

4 Tabu Search for the Weighted Tardiness Minimization in the SDST-JSP

We use here a conventional TS algorithm [4], the particular implementation is borrowed from [5]. In the next subsections we describe in detail the components of the algorithm that has been adapted to the SDST-JSP with weighted tardiness minimization, namely the neighborhood structure and the procedure for weighted tardiness estimation after a move.

4.1 The Neighborhood Structure

As usual, this structure is based on changing processing orders in a critical block. However, the number of critical paths in a problem instance is usually large and so not all candidate moves can be considered in order to obtain a reasonable number of neighbors. To do that, we consider two options: all critical paths corresponding to tardy jobs or just the critical path that most contributes to the objective function. We have observed that neither option is clearly better than the other in some preliminary experiments. Moreover, it happens that there are substantial differences that depend on the instances, so we have finally opted to consider them both and choose randomly each time the TS algorithm is issued.

Also, we have opted to consider a neighborhood structure that generates a small number of neighbors from each critical block. For this reason, we adapted the structure N_1^S proposed in [13] for SDST-JSP with makespan minimization, which is based on previous structures given in [8] and [12] for the standard JSP. This structure can be formalized for the SDST-JSP with weighted tardiness minimization from the disjunctive model defined in [2.1]. N_1^S is based on the following results.

Proposition 1. *Let H be a schedule and (v, w) a disjunctive arc that is not in a critical block. Then, reversing the arc (v, w) does not produce any improvement, provided that the triangular inequality for the setup times holds for all operations requiring the same machine.*

So we have to reverse a critical arc to obtain an improving schedule. In [13] the authors define non-improving conditions for some reversals of critical arcs in makespan optimization that in principle can not be translated to the weighted tardiness case. Regarding feasibility, the next result gives a sufficient condition for an alternative path not existing after the reversal of a critical arc. If such an alternative path exists then the resulting neighbor would be unfeasible because it would contain a cycle.

Proposition 2. *Let H be a schedule and (v, w) an arc in a critical block. A sufficient condition for an alternative path between v and w not existing is that*

$$r_{PJ_w} < r_{SJ_v} + p_{SJ_v} + \min\{S_{kl} | (k, l) \in E, J_k = J_v\}$$

where J_k is the job of operation k .

So, the neighborhood structure N_1^S is defined as follows.

Definition 1. (N_1^S) *Given a schedule H , the neighborhood $N_1^S(H)$ consists of all schedules derived from H by reversing one arc (v, w) of a critical block, provided that feasibility condition given in proposition 2 holds.*

4.2 Weighted Tardiness Estimation

Even though computing the weighted tardiness of a neighbor only requires to recompute heads (tails) of operations that are after (before) the first (last) operation moved, for the sake of efficiency the selection rule is based on weighted tardiness estimations instead of computing the actual weighted tardiness of all neighbors. For this purpose, we have extended the procedure *lpath* given for the JSP in [11] to cope with both setup times and weighted tardiness. This procedure is termed *lpathSWT* and it is shown in Algorithm 1.

Remember that each task t has N tails denoted by $q_t^1 \dots q_t^N$. For each $i = 1 \dots N$, *lpathSWT* estimates the cost of the longest path from node *start* to each node end_i through the node v or the node w , and then estimates the weighted tardiness of the neighboring schedule from the estimations of these paths. It's easy to prove that *lpathSWT* produces a lower bound for the weighted tardiness when using N_1^S .

The makespan estimation algorithm *lpath* is very accurate and very efficient. However, *lpathSWT* is much more time consuming as it calculates N tails for each operation. Moreover, experiments conclude that weighted tardiness estimation is much less accurate than makespan estimation. We have conducted a series of experiments in several instances, generating 3 millions of neighbors for each instance, and for 81.56% of neighbors the makespan estimation coincided

```

Require: A sequence of operations  $(w, v)$  as they appear after a move
Ensure: A estimation of the weighted tardiness of the resulting schedule
    TotalEst = 0;
     $r'_w = \max \{r_{PJ_w} + p_{PJ_w}, r_{PM_w} + p_{PM_w} + S_{PM_w w}\};$ 
     $r'_v = \max \{r_{PJ_v} + p_{PJ_v}, r'_w + p_w + S_{wv}\};$ 
    for  $i = 1$  to  $N$  do
         $q_v^i = \max \{q_{SJ_v}^i + p_{SJ_v}, q_{SM_v}^i + p_{SM_v} + S_{vSM_v}\};$ 
         $q_w^i = \max \{q_{SJ_w}^i + p_{SJ_w}, q_v^i + p_v + S_{wv}\};$ 
        PartialEst =  $\max \{r'_w + p_w + q_w^i, \{r'_v + p_v + q_v^i\}\};$ 
        TotalEst = TotalEst +  $(\max((PartialEst - d_i), 0) * w_i);$ 
    return TotalEst;
    
```

Alg. 1. Procedure *lpathSWT*

with the actual value, but for weighted tardiness this value drops to 51.37% of the cases.

Other authors, for example Mati et al. in [7] or Essafi et al. in [3] opted to use more accurate estimations (they report results with exact estimations from 57% to 76%, depending on the particular instance). However, their estimation procedure is more time consuming as the complexity goes up from $O(1)$ to $O(N)$ for each path, where N is the number of jobs.

For these reasons, we have opted to evaluate the actual weighted tardiness when the neighbor’s estimation is lower than the actual weighted tardiness of the original schedule. So, the use of *lpathSWT* allows the algorithm to discard a lot of neighbors in a very fast manner. Some preliminary results have shown that the improvement achieved in this way makes up the time consumed by far.

5 Experimental Study

The purpose of the experimental study is to compare $GA + TS$ with other state-of-the-art algorithms. Firstly, we compare our algorithm with the *GLS* algorithm proposed in [3] and the *MDL* algorithm proposed in [7] to solve the JSP. We experimented across the 22 instances of size 10×10 proposed by Singer and Pinedo in [9] (ABZ5, ABZ6, LA16 to LA24, MT10, and ORB01 to ORB10). Weights and due dates are defined as follows: the first 20% of the jobs have a weight 4 (very important jobs), the next 60% have weight 2 (moderately important jobs) and the remaining jobs have weight 1 (not important jobs). The due date d_i for each job i is defined in this way:

$$d_i = f * \sum_{j=1}^M p_{ij},$$

where f is a parameter that controls the tightness of the due dates. In this benchmark three values are considered: $f = 1.3$, $f = 1.5$ and $f = 1.6$.

The algorithm proposed by Essafi et al. is implemented in C++ and the experiments are carried out in a PC with a 2.8 GHz processor and 512 MB RAM,

giving a maximum runtime of 18 seconds per run. The local search proposed by Mati et al. runs in a Pentium with a 2.6 GHz processor, and they use a maximum runtime of 18 seconds too. *GA+TS* runs in a Windows XP in a Intel Core 2 Duo at 2.66GHz with 2Gb RAM. We choose the parameters /58/70/50/ (/GA population/GA generations/maximum number of iterations without improvement for TS/) for *GA + TS* to obtain a similar runtime.

Table 1 summarizes the results of these experiments; 10 trials were done for each instance and the average weighted tardiness of the 10 solutions and the number of times that the best known solution (BKS) was found are reported, “-” indicates that BKS is reached in all 10 trials.

Table 1. Results from *GLS*, *MDL* and *GA+TS* across Singer and Pinedo’s instances

Inst.	$f = 1.3$				$f = 1.5$				$f = 1.6$			
	BKS	GLS	MDL	GA+TS	BKS	GLS	MDL	GA+TS	BKS	GLS	MDL	GA+TS
ABZ5	1403	-	1414(2)	1412(7)	69	-	-	-	0	-	-	-
ABZ6	436	-	-	-	0	-	-	-	0	-	-	-
LA16	1169	1175(9)	-	-	166	-	-	166(9)	0	-	-	-
LA17	899	-	-	-	260	-	-	-	65	-	-	-
LA18	929	933(6)	934(6)	-	34	-	-	-	0	-	-	-
LA19	948	949(8)	-	998(4)	21	-	-	-	0	-	-	-
LA20	805	-	-	834(3)	0	-	-	0(7)	0	-	-	-
LA21	463	-	-	-	0	-	-	-	0	-	-	-
LA22	1064	1087(1)	1077(4)	1079(3)	196	-	-	-	0	-	-	-
LA23	835	865(2)	865(2)	870(1)	2	-	-	-	0	-	-	-
LA24	835	-	-	-	82	86(3)	86(2)	88(1)	0	-	-	-
MT10	1363	1372(9)	-	1383(9)	394	-	-	-	141	162(1)	152(1)	145(6)
ORB1	2568	2651(0)	2639(3)	2578(8)	1098	1159(6)	1247(0)	-	566	688(0)	653(0)	592(2)
ORB2	1408	1444(2)	1426(3)	1426(3)	292	-	-	-	44	-	-	-
ORB3	2111	2170(4)	2158(1)	2160(6)	918	943(4)	961(0)	939(4)	422	514(1)	463(4)	434(7)
ORB4	1623	1643(7)	1690(2)	1632(6)	358	394(8)	435(4)	-	66	78(8)	68(8)	-
ORB5	1593	1659(1)	1775(0)	1615(7)	405	-	415(8)	428(7)	163	181(0)	176(3)	176(3)
ORB6	1790	-	1793(9)	1854(5)	426	440(5)	437(5)	430(8)	28	-	-	-
ORB7	590	592(9)	-	-	50	55(8)	-	-	0	-	-	-
ORB8	2429	2522(0)	2523(0)	2477(4)	1023	1059(7)	1036(6)	1033(2)	621	669(0)	643(1)	639(3)
ORB9	1316	-	-	-	297	311(7)	299(9)	302(9)	66	83(7)	80(4)	-
ORB10	1679	1718(5)	1774(1)	1731(6)	346	400(4)	436(0)	430(2)	76	142(0)	117(0)	82(5)

GA + TS was the only algorithm capable of reaching the BKS in at least one run for all 66 instances. Globally, *GA + TS* obtains the BKS in 517 of the total 660 runs (78.3%), GLS in 443 (67.1%) and MDL in 458 (69.4%). Regarding the average weighted tardiness, we have run two t-tests with alpha level at 0.05 to compare *GA + TS* against GLS and MDL. With p-values of 0.016 and 0.010 respectively, both tests showed that there is good evidence that the mean average weighted tardiness obtained by *GA + TS* is lower than the obtained by the other methods. Overall, *GA + TS* is quite competitive with the state-of-the-art algorithms in solving the classic JSP with weighted tardiness minimization.

In the second series of experiments, we compared *GA+TS* with ILOG CPLEX CP Optimizer (CP) in solving the SDST-JSP across the BT-set proposed in [2]. We define due dates and weights as before. BT instances are divided in three groups depending on its size: small instances, t2-ps01 to t2-ps05, are 10×5 , medium instances, t2-ps06 to t2-ps10, are 15×5 , and large instances, t2-ps11 to t2-ps15, are 20×5 . These instances verify the triangular inequality for setup times. *GA + TS* was parameterized as /100/200/50/. CP was run setting the

option “Extended” for parameter “NoOverlapInferenceLevel” as the results in this case were slightly better. Both methods were run 30 times for each instance. Table (2) summarizes the results of these experiments: for each method and instance, the average value of the weighted tardiness is reported. The time taken by $GA + TS$ in a single run is given in the last column. CP was given this time plus 20% more in each run. As we can observe, $GA + TS$ reaches much better solutions than CP . On average, the weighted tardiness obtained by CP is 13.6% worse than that obtained by $GA + TS$. We have run a t-test with alpha level at 0.05, and with a p-value of 0.000 the test showed that there is strong evidence that the mean average weighted tardiness obtained by $GA + TS$ is lower than the obtained by CP .

Table 2. Results from $GA + TS$ and CP in solving SDST-JSP with weighted tardiness minimization on the BT-set

Inst.	$f = 1.3$		$f = 1.5$		$f = 1.6$		T(s)
	GA+TS	CP	GA+TS	CP	GA+TS	CP	
t2-ps01	4454	4994	3361	3911	2852	3506	47
t2-ps02	3432	4143	2674	2957	2301	2558	48
t2-ps03	4001	4609	3120	3560	2677	3143	51
t2-ps04	3732	4021	2890	3050	2539	2632	48
t2-ps05	3806	4445	2996	3532	2620	3038	45
t2-ps06	9941	10533	8238	8997	7436	8148	121
t2-ps07	9508	10552	8079	8875	7425	8642	122
t2-ps08	9902	10834	8360	9317	7624	8521	120
t2-ps09	9998	11569	8215	9697	7317	8813	124
t2-ps10	10569	11999	8745	9968	7914	8848	116
t2-ps11	23052	26169	20816	24132	19764	22909	229
t2-ps12	23158	25331	21119	22309	20106	21039	241
t2-ps13	24026	25729	21821	23548	20618	22279	244
t2-ps14	25416	27569	23051	25580	21892	24079	250
t2-ps15	25427	27144	23028	25450	22049	23919	237

6 Conclusions

Our study of SDST-JSP has demonstrated that metaheuristics such as genetic algorithms and tabu search are very efficient in solving complex scheduling problems. These techniques are flexible and robust, so as they can be adapted to the particular characteristics of a given problem. Here, we have seen how a solution designed to cope with makespan can be adapted to cope with weighted tardiness, which is well-known that is harder to optimize. Also, we have demonstrated that for weighted tardiness minimization, a specific solution based on specific knowledge of the problem domain can be much more efficient than a solution built on a general purpose solver.

As future work, we plan to consider other scheduling problems and different objective functions, even non-regular objective functions, i.e. objective functions that could be decreasing on the completion times of the operations, such as robustness or stability measures.

Acknowledgments. This research has been supported by the Spanish Ministry of Science and Innovation under research project MICINN-FEDER TIN2010-20976-C02-02 and by the Principality of Asturias under grant FICYT-BP07-109.

References

1. Artigues, C., Lopez, P., Ayache, P.D.: Schedule generation schemes for the job shop problem with sequence-dependent setup times: Dominance properties and computational analysis. *Annals of Operations Research* 138, 21–52 (2005)
2. Brucker, P., Thiele, O.: A branch and bound method for the general-job shop problem with sequence-dependent setup times. *Operations Research Spektrum* 18, 145–161 (1996)
3. Essafi, I., Mati, Y., Dauzère-Pérès, S.: A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem. *Computers and Operations Research* 35, 2599–2616 (2008)
4. Glover, F., Laguna, M.: *Tabu Search*. Kluwer Academic Publishers (1997)
5. González, M.A., Vela, C.R., Varela, R.: Genetic Algorithm Combined with Tabu Search for the Job Shop Scheduling Problem with Setup Times. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) *IWINAC 2009*. LNCS, vol. 5601, pp. 265–274. Springer, Heidelberg (2009)
6. González, M.A., Vela, C., Varela, R.: A new hybrid genetic algorithm for the job shop scheduling problem with setup times. In: *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS 2008)*, pp. 116–123. AAAI Press, Sidney (2008)
7. Mati, Y., Dauzere-Peres, S., Lahlou, C.: A general approach for optimizing regular criteria in the job-shop scheduling problem. *European Journal of Operational Research* (2011), doi:10.1016/j.ejor.2011.01.046
8. Matsuo, H., Suh, C., Sullivan, R.: A controlled search simulated annealing method for the general jobshop scheduling problem. Working paper 03-44-88, Graduate School of Business, University of Texas (1988)
9. Singer, M., Pinedo, M.: A computational study of branch and bound techniques for minimizing the total weighted tardiness in job shops. *IIE Transactions* 30, 109–118 (1998)
10. Sun, X., Noble, J.: An approach to job shop scheduling with sequence-dependent setups. *Journal of Manufacturing Systems* 18(6), 416–430 (1999)
11. Taillard, E.: Parallel taboo search techniques for the job shop scheduling problem. *ORSA Journal on Computing* 6, 108–117 (1993)
12. Van Laarhoven, P., Aarts, E., Lenstra, K.: Job shop scheduling by simulated annealing. *Operations Research* 40, 113–125 (1992)
13. Vela, C.R., Varela, R., González, M.A.: Local search and genetic algorithm for the job shop scheduling problem with sequence dependent setup times. *Journal of Heuristics* 16, 139–165 (2010)

Interval-Valued Fuzzy Sets for Color Image Super-Resolution

Aranzazu Jurio, José Antonio Sanz, Daniel Paternain,
Javier Fernandez, and Humberto Bustince

Universidad Publica de Navarra, 31006 Pamplona, Spain
aranzazu.jurio@unavarra.es

Abstract. In this work we associate an Interval-Valued Fuzzy Set with an image, so that the membership of each pixel represents the intensities of itself and its neighbourhood. Based on this set we propose a new simple super-resolution algorithm for color images. We show some experimental results and study how the δ parameter has influence on the results obtained by the algorithm.

1 Introduction

Interval-valued fuzzy theory has been widely used in image processing because it allows to keep information about the neighbourhood of each pixel. It has been used to solve problems like edge detection [3], filtering [2] or segmentation [15], [4], [7]. In this work, we present a new method to associate an interval-valued fuzzy set (IVFS) to an image. The interval membership of each pixel represents its original intensity and its neighbourhoods' one, being the length of that membership a measure of the variation of intensities in the neighbourhood of that pixel.

We propose a new super-resolution algorithm. Image super-resolution (or magnification) [16], [13], [10], [11] has a wide range of application nowadays. For instance, the uploading of images to a web page or the display of images in devices such as mobile phones, PDAs or screens. The memory of some of these devices is very limited, so the need to use simple image magnification algorithms.

Many techniques for image magnification can be found in the literature. They are based on one image or on several ones. Among the methods working with a single image, the most frequent ones are based on interpolation [1], [8]. Common algorithms such as nearest neighbour or bilinear interpolation are computationally simple, but they suffer from smudge problems, especially in the areas containing edges. Nevertheless, linear approximations are the most popular ones based on their low computational cost, even providing worse results than cubic interpolation or splines.

The main application of methods working with several images is the magnification of video sequences [12], [14], although they are also used to enlarge individual images in learning frameworks [6], [9].

Our approach is based on a single image, where the magnified image is obtained by joining different constructed blocks. To create each block we use the

interval-valued fuzzy set that we have previously associated with the image, maintaining the intensity of the original pixel in the center of the block and filling in the rest using the relation between that pixel and its neighbours.

This work is organized as follows: in Section 2 we recall some preliminary definitions. In Section 3 we show the construction method of IVFSs. The super-resolution algorithm is described in detail in Section 4. We finish this work with some illustrative examples in Section 5 and conclusions in Section 6.

2 Preliminaries

Let us denote by $L([0, 1])$ the set of all closed subintervals in $[0, 1]$, that is,

$$L([0, 1]) = \{\mathbf{x} = [\underline{x}, \bar{x}] \mid (\underline{x}, \bar{x}) \in [0, 1]^2 \text{ and } \underline{x} \leq \bar{x}\}.$$

$L([0, 1])$ is a lattice with respect to the relation \leq_L , which is defined in the following way. Given $\mathbf{x}, \mathbf{y} \in L([0, 1])$,

$$\mathbf{x} \leq_L \mathbf{y} \text{ if and only if } \underline{x} \leq \underline{y} \text{ and } \bar{x} \leq \bar{y}.$$

The relation above is transitive, antisymmetric and it expresses the fact that \mathbf{x} strongly links to \mathbf{y} , so that $(L([0, 1]), \leq_L)$ is a complete lattice, where the smallest element is $0_L = [0, 0]$, and the largest is $1_L = [1, 1]$.

Definition 1. *An interval-valued fuzzy set A on the universe $U \neq \emptyset$ is a mapping $A : U \rightarrow L([0, 1])$.*

We denote by $IVFSs(U)$ the set of all IVFSs on U . Similarly, $FSSs(U)$ is the set of all fuzzy sets on U .

From now on, we denote by $W([\underline{x}, \bar{x}])$ the length of the interval $[\underline{x}, \bar{x}]$; that is, $W([\underline{x}, \bar{x}]) = \bar{x} - \underline{x}$.

Definition 2. *Let $\alpha \in [0, 1]$. The operator $K_\alpha : L([0, 1]) \rightarrow [0, 1]$ is defined as a convex combination of the bounds of its argument, i.e.*

$$K_\alpha(\mathbf{x}) = \underline{x} + \alpha(\bar{x} - \underline{x})$$

for all $\mathbf{x} \in L([0, 1])$.

Clearly, the following properties hold:

1. $K_0(\mathbf{x}) = \underline{x}$ for all $\mathbf{x} \in L([0, 1])$,
2. $K_1(\mathbf{x}) = \bar{x}$ for all $\mathbf{x} \in L([0, 1])$,
3. $K_\alpha(\mathbf{x}) = K_\alpha([K_0(\mathbf{x}), K_1(\mathbf{x})]) = K_0(\mathbf{x}) + \alpha(K_1(\mathbf{x}) - K_0(\mathbf{x}))$ for all $\mathbf{x} \in L([0, 1])$.

Let $A \in IVFSs(U)$ and $\alpha \in [0, 1]$. Then, we denote by $K_\alpha(A)$ the fuzzy set

$$K_\alpha(A) = \{u_i, K_\alpha(A(u_i)) \mid u_i \in U\}.$$

Proposition 1. *For all $\alpha, \beta \in [0, 1]$ and $A, B \in IVFSs(U)$, it is verified that*

- (a) *If $\alpha \leq \beta$, then $K_\alpha(A) \leq K_\beta(A)$.*
- (b) *If $A \leq_L B$ then $K_\alpha(A) \leq K_\alpha(B)$.*

3 Construction of Interval-Valued Fuzzy Sets

In this section we propose a method to associate an image with an IVFS. We demand two properties to this method: the first one is that the intensity of every pixel in the original image must belong to the interval membership associated with it. The second one is that the length of each interval membership must depend on the intensities of the original pixel and its neighbours'. In this sense, we represent the variation of the intensities around each pixel, adjusted by a scaling factor (δ), by the length of the interval.

Proposition 2. *The mapping $F : [0, 1]^2 \times [0, 1] \rightarrow L([0, 1])$ given by*

$$F(x, y, \delta) = [\underline{F}(x, y, \delta), \overline{F}(x, y, \delta)]$$

where

$$\begin{aligned} \underline{F}(x, y, \delta) &= x(1 - \delta y) \\ \overline{F}(x, y, \delta) &= x(1 - \delta y) + \delta y \end{aligned}$$

satisfies that:

1. $\underline{F}(x, y, \delta) \leq x \leq \overline{F}(x, y, \delta)$ for all $x \in [0, 1]$;
2. $\overline{F}(x, 0, \delta) = [x, x]$;
3. $\overline{F}(0, y, \delta) = [0, \delta y]$;
4. $\underline{F}(x, y, 0) = [x, x]$;
5. $W(\underline{F}(x, y, \delta)) = \delta y$.
6. If $y_1 \leq y_2$ then $W(\underline{F}(x, y_1, \delta)) \leq W(\underline{F}(x, y_2, \delta))$ for all $x, \delta \in [0, 1]$;

Theorem 1. *Let $A_F \in FSS(U)$ and let $\omega, \delta : U \rightarrow [0, 1]$ be two mappings. Then*

$$A = \{(u_i, A(u_i) = F(\mu_{A_F}(u_i), \omega(u_i), \delta(u_i))) | u_i \in U\}$$

is an Interval-Valued Fuzzy Set.

Corollary 1. *In the setting of Theorem 1, if for every $u_i \in U$ we take $\delta(u_i) = 1$ then*

$$\omega(u_i) = W(F(\mu_{A_F}(u_i), \omega(u_i), 1)).$$

Notice that under the conditions of Corollary 1 the set A is given as follows:

$$A = \{(u_i, \mu_{A_F}(u_i)(1 - \omega(u_i)), \mu_{A_F}(u_i)(1 - \omega(u_i)) + \omega(u_i)) | u_i \in U\}$$

Example 1. Let $U = \{u_1, u_2, u_3, u_4\}$ and let $A_F \in FSS(U)$ given by

$$A_F = \{(u_1, 0.3), (u_2, 1), (u_3, 0.5), (u_4, 0.8)\}$$

and $\omega(u_i) = 0.3, \delta(u_i) = 1$ for all $u_i \in U$. By Corollary 1 we obtain the following Interval-Valued Fuzzy Set:

$$A = \{(u_1, [0.21, 0.51]), (u_2, [0.7, 1.00]), (u_3, [0.35, 0.65]), (u_4, [0.56, 0.86])\}$$

4 Super-Resolution Algorithm

In this section we propose a color image super-resolution algorithm based on block expansion, that uses IVFSs and the K_α operators.

We consider a color image Q in the RGB space as a $N \times M \times 3$ matrix. Each coordinate of the pixels in the image Q is denoted by (i, j, k) . The normalized intensity of the pixel located at (i, j, k) is represented as q_{ijk} , with $0 \leq q_{ijk} \leq 1$ for each $(i, j, k) \in Q$.

The purpose of our algorithm is, given an image Q of dimension $N \times M \times 3$, to magnify it $n \times m$ times; that is, to build a new image of dimension $N' \times M' \times 3$ with $N' = n \times N$, $M' = m \times M$, $n, m \in \mathbb{N}$ with $n \leq N$ and $m \leq M$. We denote $(n \times m)$ as magnification factor.

INPUT: Q original image, $(n \times m)$ magnification factor.

1. Take $\delta \in [0, 1]$.
2. FOR each pixel in each channel (i, j, k) DO
 - 2.1. Fix a grid V of dimension $n \times m \times 1$ centered at (i, j, k) .
 - 2.2. Calculate W as the difference between the largest and the smallest intensities of the pixels in V .
 - 2.3. Build the interval $F(q_{ijk}, W, \delta)$.
 - 2.4. Build a block V' equal to V .
 - 2.5. FOR each element (r, s) of V' DO

$$q_{rs} := K_{qrs}(F(q_{ijk}, W, \delta)).$$

ENDFOR

- 2.6. Put the block V' in the magnified image.

ENDFOR

Algorithm 1.

We explain the steps of this algorithm by means of an example. Given an image in Figure 1 of dimension $5 \times 5 \times 3$, we want to build a magnified image of dimension $15 \times 15 \times 3$ (magnification factor= (3×3)).

Step 1. Take $\delta \in [0, 1]$

In the example we take the middle value, $\delta = 0.5$.

Step 2.1. Fix a grid V of dimension $m \times n \times 1$ centered at each pixel

This grid V represents the neighborhood that is used to build the interval. In the example, for pixel $(2, 3, 1)$ (marked in dark gray in Figure 1), we fix a grid of dimension 3×3 around it (in light gray).

Step 2.2. Calculate W as the difference between the largest and the smallest of the intensities of the pixels in V

For pixel $(2, 3, 1)$, we calculate W as:

0.60	0.65	0.68	0.70	0.70
0.60	0.68	0.70	0.72	0.73
0.70	0.69	0.69	0.73	0.75
0.15	0.14	0.15	0.17	0.19
0.13	0.16	0.12	0.15	0.21

(a)

0.20	0.20	0.25	0.25	0.30
0.20	0.21	0.20	0.20	0.30
0.22	0.20	0.25	0.60	0.65
0.60	0.65	0.62	0.65	0.70
0.65	0.70	0.70	0.73	0.75

(b)

0.79	0.83	0.85	0.82	0.86
0.81	0.82	0.79	0.83	0.81
0.80	0.81	0.50	0.83	0.84
0.52	0.50	0.51	0.51	0.54
0.50	0.51	0.53	0.52	0.53

(c)

Fig. 1. Example: original color image. (a) The R channel of the image. (b) The G channel of the image. (c) The B channel of the image. In grey it is shown the grid V for pixel $(2, 3, 1)$.

$$\begin{aligned}
 W &= \max(0.65, 0.68, 0.70, 0.68, 0.70, 0.72, 0.69, 0.69, 0.73) - \\
 &\quad \min(0.65, 0.68, 0.70, 0.68, 0.70, 0.72, 0.69, 0.69, 0.73) = \\
 &= 0.73 - 0.65 = 0.08
 \end{aligned}$$

W is the maximum length of the interval associated with the pixel. The final length is calculated scaling it by the factor δ chosen in Step 1.

Step 2.3. Build interval $F(q_{ijk}, W, \delta)$

We associate to each pixel an interval of length $\delta \cdot W$ by the method explained in Section 3.

$$F(q_{ijk}, \delta \cdot W) = [q_{ijk}(1 - \delta \cdot W), q_{ijk}(1 - \delta \cdot W) + \delta \cdot W].$$

In the example, the interval associated to pixel $(2, 3, 1)$ is given by:

$$F(0.7, 0.08, 0.5) = [0.7(1 - 0.08 \cdot 0.5), 0.7(1 - 0.08 \cdot 0.5) + 0.08 \cdot 0.5] = [0.672, 0.712]$$

Step 2.4. Build a block V' equal to V

In the example, this new block is shown in Figure 2.

0.65	0.68	0.70
0.68	0.70	0.72
0.69	0.69	0.73

Fig. 2. Original V' block for pixel $(2, 3)$

Step 2.5. Calculate $K_{q_{rs}}(F(q_{ijk}, W, \delta))$ for each pixel

Next, we expand the pixel (i, j, k) in image Q over the new block V' . In the example, the pixel $(2, 3, 1)$ is expanded as shown in Figure 3.

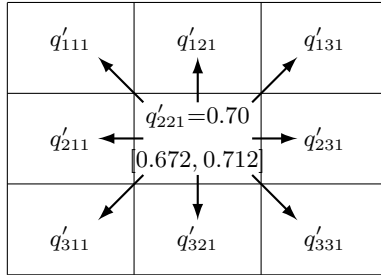


Fig. 3. Expanded block for pixel q_{231}

To keep the value of the original pixel at the center of the new block, we use the result obtained in Proposition 3.

Proposition 3. *In the settings of Proposition 1, if we take $\alpha = x$, then*

$$K_x(F(x, y, \delta)) = x$$

for all $x, y, \delta \in [0, 1]$.

Proof. $K_x(F(x, y, \delta)) = K_x([x(1-\delta y), x(1-\delta y)+\delta y]) = x(1-\delta y) + x \cdot W(F(x, y, \delta))$.

This proposition states that if we take α as the intensity of the pixel, we recover that same intensity from the constructed interval. In the case of pixel $(2, 3, 1)$ of the example we have

$$0.7 = q'_{221} = K_{q'_{221}}([0.672, 0.712]) = 0.672 + q'_{221} \cdot 0.04 = 0.7.$$

We apply this method to fill in all the other pixels in the block. In this way, from Proposition 3 we take as α for each pixel, the value of that pixel in the grid V' :

- $\alpha = q'_{111}$. Then $q'_{111} = 0.672 + 0.65 \cdot 0.04 = 0.698$
- $\alpha = q'_{121}$. Then $q'_{121} = 0.672 + 0.68 \cdot 0.04 = 0.6992$
- ...
- $\alpha = q'_{331}$. Then $q'_{331} = 0.672 + 0.73 \cdot 0.04 = 0.7012$

In Figure 4 we show the expanded block for pixel $(2, 3, 1)$ in the example.

Once each of the pixels has been expanded, we join all the blocks (Step 2.6) to create the magnified image. This process can be seen in Figure 5.

0.6980	0.6992	0.7000
0.6992	0.7000	0.7008
0.6996	0.6996	0.7012

Fig. 4. Numerical expanded block for pixel q_{231}

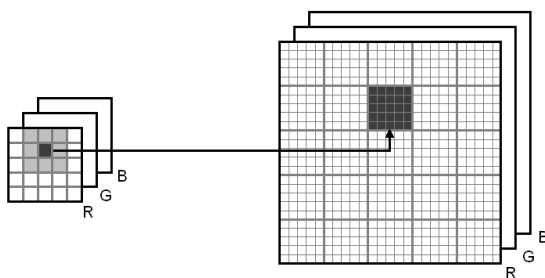


Fig. 5. Construction of the magnified image by joining all the created blocks

5 Illustrative Examples

In this section we show some illustrative examples of the proposed algorithm. To evaluate the quality of the results we start from color images of 510×510 and we reduce them to 170×170 using the reduction algorithm proposed in [4]. By means of Algorithm 1 we magnify the reduced images to a 510×510 size. Finally, we compare the obtained images with the original ones, using PSNR (see Figure 6).

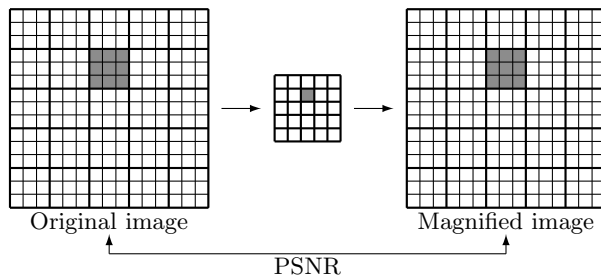


Fig. 6. Scheme to check the algorithm

Algorithm 1 has a parameter δ that is a scaling factor. This parameter can range between 0 and 1. In Figure 7 it is shown the original images Beeflower, Elephant, Frog and Safari, their reduced versions and the magnified images obtained by Algorithm 1 with $\delta = 0.5$ (middle value).

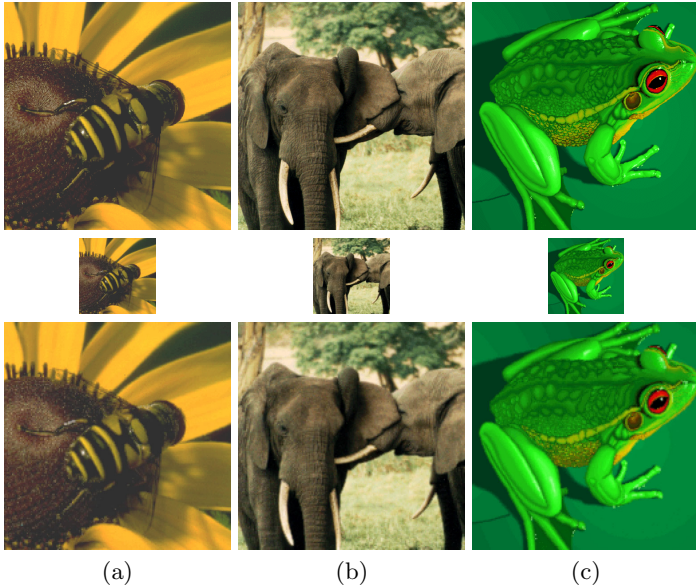


Fig. 7. First row: Original images (a) Beeflower, (b) Elephant and (c) Frog. Their reductions (second row) and magnifications with $\delta = 0.5$ (third row).

As we have said, depending on the value of δ parameter, the obtained results vary. When $\delta = 0$, we know by Proposition 2 that $F(x, y, 0) = [x, x]$ for all $x, y \in [0, 1]$, and we also know that $K_\alpha([x, x]) = x$ for all $\alpha \in [0, 1]$. In this sense, when $\alpha = 0$ we build blocks in which all the elements take the value of the central pixel, and we lose information from the neighbourhood. But when δ increases, the length of the interval associated with each pixel increases too, so the range in which the intensities of pixels in each reconstructed block vary is bigger. If we take the biggest possible value ($\delta = 1$) every block intensities vary in a too big range, so there are some blurring problems. In this sense, an intermediate δ value allows to balance these two problems: jaggling artifacts and blurring. In Figure 8 we show three cropped images magnified with five different δ values ($\delta = 0$, $\delta = 0.25$, $\delta = 0.5$, $\delta = 0.75$ and $\delta = 1$).

To compare the obtained images with the original ones we use the PSNR measure (see Table I). We observe that the same conclusion is obtained: the best results are got with intermediate values of δ parameter.

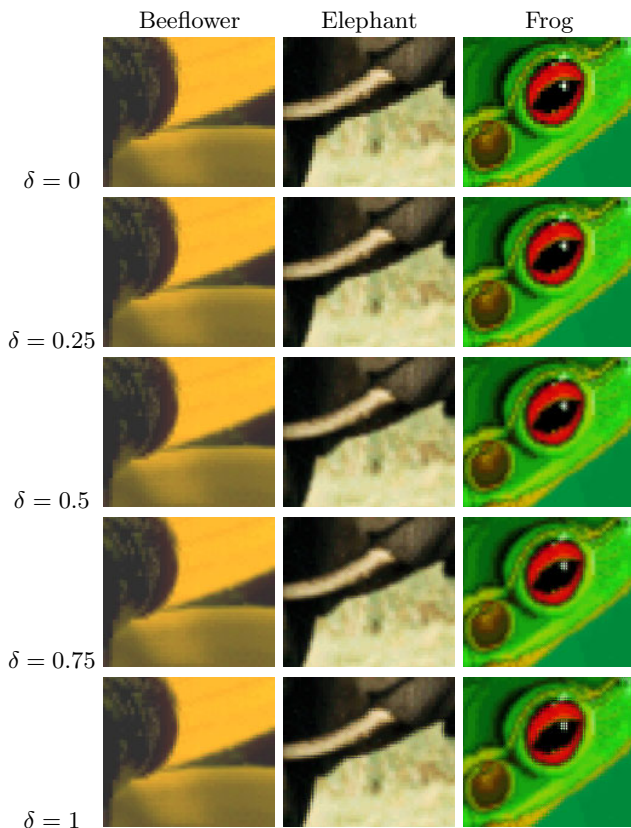


Fig. 8. Reconstructed images with different values of the parameter δ

Table 1. Comparison of the magnified images with the original one

	$\delta = 0$	$\delta = 0.25$	$\delta = 0.5$	$\delta = 0.75$	$\delta = 1$
Beeflower	29.3964	29.8177	30.0912	30.1858	30.0895
Elephant	27.3838	28.0579	28.4589	28.4995	28.1698
Frog	27.7417	28.1376	28.3393	28.3182	28.0774

6 Conclusions

In this work we have introduced a new magnification algorithm for color images. It is based on block expansion and it is characterized by its simplicity. This method uses interval-valued fuzzy sets to keep the information of the neighbourhood of each pixel. Besides, it maintains the original intensities with K_α operators. The parametrization used in the algorithm allows to adapt it in order to look for the optimal set-up for each image, balancing the solutions with jaggig artifacts and the solutions with blurring.

Acknowledgement. This paper has been partially supported by the National Science Foundation of Spain, reference TIN2010-15055 and by the Research Services of the Universidad Publica de Navarra.

References

1. Amanatiadis, A., Andreadis, I.: A survey on evaluation methods for image interpolation. *Measurement Science & Technology* 20, 104015 (2009)
2. Bigand, A., Colot, O.: Fuzzy filter based on interval-valued fuzzy sets for image filtering. *Fuzzy Sets and Systems* 161, 96–117 (2010)
3. Bustince, H., Barrenechea, E., Pagola, M., Fernandez, J.: Interval-valued fuzzy sets constructed from matrices: Application to edge detection. *Fuzzy Sets and Systems* 160, 1819–1840 (2009)
4. Bustince, H., Paternain, D., De Baets, B., Calvo, T., Fodor, J., Mesiar, R., Montero, J., Pradera, A.: Two Methods for Image Compression/Reconstruction using OWA Operators. In: Yager, R.R., Kacprzyk, J., Beliakov, G. (eds.) *Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice*. STUDEFUZZ, vol. 265, pp. 229–253. Springer, Heidelberg (2011)
5. Bustince, H., Barrenechea, E., Pagola, M., Fernandez, J., Sanz, J.: Comment on: "Image thresholding using type II fuzzy sets". Importance of this method. *Pattern Recognition* 43, 3188–3192 (2010)
6. Gajjar, P.P., Joshi, M.V.: New Learning Based Super-Resolution: Use of DWT and IGMRF Prior. *IEEE Transactions on Image Processing* 19, 1201–1213 (2010)
7. Jurio, A., Pagola, M., Paternain, D., Lopez-Molina, C., Melo-Pinto, P.: Interval-valued restricted equivalence functions applied on Clustering Techniques. In: *13th International Fuzzy Systems Association World Congress and 6th European Society for Fuzzy Logic and Technology Conference 2009, Portugal* (2009)
8. Karabassis, E., Spetsakis, M.E.: An analysis of image interpolation, defferentiation, and reduction using local polynomial. *Graphical Models and Image Processing* 57, 183–196 (1995)
9. Ni, K.S., Nguyen, T.Q.: Image Superresolution Using Support Vector Regression. *IEEE Transactions on Image Processing* 16, 1596–1610 (2007)
10. Park, S.C., Park, M.K., Kang, M.G.: Super-resolution image reconstruction: A technical overview. *IEEE Signal Processing Magazine* 20, 21–36 (2003)
11. Perfilieva, I.: Fuzzy transforms: Theory and applications. *Fuzzy Sets and Systems* 157, 993–1023 (2006)
12. Protter, M., Elad, M.: Super Resolution With Probabilistic Motion Estimation. *IEEE Transactions on Image Processing* 18, 1899–1904 (2009)
13. Qiu, G.: Interresolution Look-up Table for Improved Spatial Magnification of Image. *Journal of Visual Communication and Image Representation* 11, 360–373 (2000)
14. Takeda, H., Milanfar, P., Protter, M., Elad, M.: Super-Resolution Without Explicit Subpixel Motion Estimation. *IEEE Transactions on Image Processing* 18, 1958–1975 (2009)
15. Tizhoosh, H.R.: Image thresholding using type II fuzzy sets. *Pattern Recognition* 38, 2363–2372 (2005)
16. Unser, M., Aldroubi, A., Eden, M.: Enlargement and reduction of digital images with minimum loss of information. *IEEE Transactions on Image Processing* 4, 247–258 (1995)

An Evolutionary Multiobjective Constrained Optimisation Approach for Case Selection: Evaluation in a Medical Problem^{*}

Eduardo Lupiani, Fernando Jimenez, José M. Juarez, and José Palma

Computer Science Faculty, Universidad de Murcia, Spain
{elupiani, fernan, jmjuarez, jtpalma}@um.es

Abstract. A solid building process and a good evaluation of the knowledge base are essential in the clinical application of Case-Based Reasoning Systems. Unlike other approaches, each piece of the knowledge base (cases of the case memory) is knowledge-complete and independent from the rest. Therefore, the main issue to build a case memory is to select which cases must be included or removed. Literature provides a wealth of methods based on instance selection from a database. However, it can be also understood as a multiobjective problem, maximising the accuracy of the system and minimising the number of cases in the case memory. Most of the efforts done in this evaluation of case selection methods focus on the number of registers selected, providing an evaluation of the system based on its accuracy. On the one hand, some case selection methods follow a non deterministic approach. Therefore, a rough evaluation could entail to inaccurate conclusions. On the other hand, specificity and sensitivity are critical values to evaluate tests in the medical field. However, these parameters are hardly ever included in the case selection evaluation. In order to partially solve this problem, we propose an evaluation methodology to obtain the best case selection method for a given memory case. We also propose a case selection method based on multiobjective constrained optimisation for which Evolutionary Algorithms are used. Finally, we illustrate the use of this methodology by evaluating classic and the case selection method proposed, in a particular problem of Burn Intensive Care Units.

1 Introduction

Case-Based Reasoning Systems (CBRS) requires a solid building process and a good evaluation of the case memory. Building processes mainly focus on defining (1) how the cases (instances) are stored in the case memory, (2) a retrieval strategy to obtain the most similar cases from the case memory and (3) how the solution is built from these knowledge [11]. The evaluation of knowledge-based systems, including CBRS, depends on their knowledge base. Unlike other approaches, such as rule-based or model-based systems [16], in a CBRS each piece of the knowledge base (cases of the case memory) is knowledge-complete and independent from the rest. Therefore, the main issue to build

^{*} This study was partially financed by the Spanish MEC through projects TIN2009-14372-C03-01 and PET2007-0033 and the project 15277/PI/10 funded by Seneca Agency of Science and Technology of the Region of Murcia within the II PCTRM 2007-2010.

a case memory is to select which cases must be included or removed. During the building process, the case memory can be obtained by selecting registers from an external database. The automatic process to carry out this selection is known as instance selection, case selection or case mining [15][17].

The present work is aimed to evaluate case selection methods. We briefly review what kind of case selection families and which evaluation techniques have been studied in literature.

Case selection methods can be classified either by the case selection methodology [23] or by case memory construction technique [17]. Among the case selection methods there are four outstanding families: (i) nearest neighbour editing rules [8][19][22], (ii) instance-based [1][2][23], (iii) prototype-based [4] and (iv) competence-based [15][20]. Both (i) and (ii) select a case as candidate to final case memory in basis to KNN classification and (iii) could additionally adapt cases or modify them. Specially interesting methods are those based on competence techniques (iv) since they introduce new concepts such as coverage and reachability; however they need good understanding of domain problem. Case selection and feature selection approaches have been very popular research issues to reduce noise in case memory and enhance system response times respectively, however almost all studies face them separately. Multiobjective evolutionary algorithms (MOEA) makes possible to cope both objectives at the same time [3][12]. Nevertheless, most experiments fall into evolutionary algorithms test sets and they are not based in popular MOEA implementations such as NSGA-II [6] and SPEA2 [24].

The Hold-Out validation is a suitable approach to evaluate the case selection methods due to large size of initial database. Experiments as [14][17][19] uses Hold-Out repeatedly a determined number of times. Furthermore, some studies try to analyse the effects of case memory size in the classifier accuracy. However, these studies are restricted to KNN based classifiers. In these cases, the case selection process finishes when a concrete (and pre-established) size of the case memory is reached [14][17]. Although Hold-Out is appropriate for large cases sets, it could present a high variance. Therefore, some researchers prefer Cross-Validation [23]. Despite the impact of this process in the CBRS evaluation, some improvements can be done in this direction.

This work focuses on the case selection task and its evaluation in medical databases. To this end, we introduce an evaluation method for case selection algorithms. In order to demonstrate the usefulness of this methodology, we present new case selection algorithms based on evolutionary multiobjective constrained optimisation. We compare the classical algorithms and the evolutionary multiobjective constrained optimisation approach in order to select the most suitable case selection algorithm according to different standard problems and a real problem in the Burn Intensive Care Unit domain.

The structure of this work is as follows. In Section 2 we propose an evaluation methodology for case selection methods. Section 3 describes an evolutionary multiobjective constrained optimisation approach for case selection. In Section 4 we evaluate the case selection method proposed in a real problem of the Burn Intensive Care Unit domain. Finally, conclusions and future works are described in Section 5.

2 Evaluation Methodology

2.1 Notation

Let C be the universe of all possible cases in a give domain and $c \in C$ be a particular case. We assume that a case is a vector of n attributes $c = (c_1, \dots, c_n)$, where each attribute could describe a quantitative or qualitative value, and with c_n represents a class or solution.

Let also define $\mathbb{M} = \wp(C)$ as the space of possible case memories and $M \in \mathbb{M}$ a particular case memory. Therefore, a case c of M is denoted by $c \in M$.

Let $\{M_i, i \in \{1, \dots, f\}\}$ be a partition of M ($\bigcup_{i=1}^f M_i = M, \forall i, j M_i \cap M_j = \emptyset$). We define the complement of M_i (element of a partition of M) as $\overline{M_i} = M \setminus M_i$.

A case selection method can be defined by the function σ as follows:

$$\sigma : \mathbb{M} \rightarrow \mathbb{M} \quad (1)$$

where $|\sigma(M)| \leq |M|$. For the sake of clarity, we denote M^σ as $\sigma(M)$. In Section 3 an example of implementation of σ is defined based on evolutionary multiobjective optimisation. Given M^σ we define the reduction rate as:

$$\rho : \mathbb{M}^\sigma \rightarrow \{0, 1\} \quad (2)$$

$$\rho(M^\sigma) = \frac{|M| - |M^\sigma|}{|M|} \quad (3)$$

Therefore, $\rho(M^\sigma) = 0$ means that σ makes no reduction, and $\rho(M^\sigma) = 1$ implies that no cases were selected from M using the σ function.

2.2 Evaluation Methodology

The final goal of the evaluation methodology is to obtain the best case selection method for a given memory case for solving a particular problem. According to this vague definition, we assume 3 main dimensions: the size of the case memory after the selection process, the efficiency of the method, and the suitability of the case memory to solve the problem.

In order to analyse these aspects we propose a 4-step methodology to evaluate case selection algorithms:

1. Calculate $\{M_i, i \in \{1 \dots f\}\}$, a randomly partition of M and $\overline{M_i}$ the complement of each element M_i .
2. For each $\overline{M_i}$ repeat α times:
 - (a) Apply the control and the case selection method ($\overline{M_i}^\sigma$).
 - (b) Validate the classifier using Cross-Validation where $\overline{M_i}^\sigma$ is the training set and M_i the test set.
 - (c) Calculate: reduction of the case memory, efficiency of the method, and quality of the solution.
3. Calculate the decision scores: average the calculi obtained from step 2.c.

In the first step, the partition is made to identify the test and training sets. In step 2.a the case selection method is applied in order to reduce the case memory. Note that the case selection method selected should not produce an adverse effect on the CBRS. In order to obtain an initial filtering of the case selection methods, we could compare them with a control test. In our case, these control methods are: the random selection process (removing 25%, 50% or 75% of the cases from the case memory) and the *none* selection (keeping the original case memory). Therefore, this methodology only considers acceptable those case selection methods whose results improve or keep the control methods.

The step 2.b is a classical Cross-Validation process. Due to the fact that case selection methods are used to improve CBRSs, it seems reasonable to include the own CBRS at this step. However, this kind of systems (such as a CBR) could imply high computationally-cost processes (e.g. similarity or adaptation functions) and the validation step implies a high number of iterations. Therefore, the custom cross-validation presented (folder size f) executes a case selection method using the training set \overline{M}_i^σ , the test set M_i , and the KNN as classifier (iterating over $i = 1, \dots, f$). The KNN has two components: local and global distances, where the global depends on local. The local is the distance between the case attributes values, therefore its calculation depends on the attribute type. In our evaluation there are just two types: numeric and string of characters, and we call d_{num} the distance between numeric values and d_{string} the distance between string values:

$$d_{num}(c_i, c'_i) = \frac{|c_i - c'_i|}{|max(c_i) - max(c'_i)|} \tag{4}$$

$$d_{string}(c_i, c'_i) = \begin{cases} 1 & \text{if } c_i \neq c'_i \\ 0 & \text{if } c_i = c'_i \end{cases} \tag{5}$$

where $max(c_i)$ and $max(c'_i)$ are the maximum domain values for the attributes c_i and c'_i .

Given two particular cases $c = (c_1, \dots, c_n)$ and $c' = (c'_1, \dots, c'_n)$ with the same number of attributes, the global euclidean distance is defined as:

$$D(c, c') = \sqrt{\sum_{i=1}^n d_i(c_i, c'_i)^2} \tag{6}$$

where d_i is d_{num} or d_{string} depending on the nature of the i -th attribute.

In the step 2.c of the methodology proposed the reduction of the case memory is evaluated by the reduction rate function ρ (expression 2). The efficiency is calculated by the average of the execution time of the reduction process. We consider the generalised error rate to evaluate the classification, the κ coefficient to evaluate the coincidence of the solution and specificity and sensitivity values to analyse particular problems of the domain application.

Finally, in the step 3 the evaluation scores are calculated averaging the results obtained in the step 2.c since they have been calculated α times. Note that some of

the case selection techniques are non deterministic algorithm (e.g. evolutionary algorithms), therefore, in order to obtain a correct evaluation, the case selection must be performed α times ($\alpha = 100$ is usually accepted in the field).

3 An Evolutionary Multiobjective Constrained Optimisation Approach for Case Selection

3.1 A Multiobjective Constrained Optimisation Model for Case Selection

The selection of cases concerns finding the smallest subset of cases in a database to obtain the most accurate classification possible. Described more formally, lets suppose an initial case memory M where $|M| = X$, the algorithm finds $M^\sigma \subseteq M$, removing the irrelevant or redundant cases, and obtaining good accuracy of the classification. For the sake of clarity, since the algorithm could obtain different M^σ sets, we denote them by x, y, z , etc.

Therefore, as in [9] for attribute selection, the problem of cases selection can be approached as a multiobjective optimisation problem, the solution of which comprise as set of solutions called non-dominated solutions (or Pareto solutions). Given two solutions $x = \{c|c \in M\}$ and $y = \{c|c \in M\}$, solution x dominates solution y if [6]:

- Solution x is not worse than y for any of the objectives.
- Solution x is strictly better than y for at least one of the objectives.

For the case selection problem in mind, two optimisation criteria, *accuracy* and *compactness*, and a constraint, *coverage*, have been considered. To formulate these criteria and the constraint, the following quantitative measures have been defined.

Given a solution x of M , we define:

- Accuracy: based on the error ratio $ER(x) = \frac{\Phi(x)}{|x|}$, where $\Phi(x)$ is the number of cases misclassified for a set of cases, x , by a given classification algorithm.
- Compactness: by cardinality $|x|$, that is, the number of cases used to construct a classification model.
- Coverage: the set of cases x requires that all the different classes or solutions of M be covered at least for one case, i.e., $CV(x) = CV(M)$, where $CV(M)$ is the number of different classes or solutions covered by a case memory M .

According to these criteria and the constraint, we propose the following optimisation model:

$$\begin{aligned}
 & \text{Minimise } ER(x) \\
 & \text{Minimise } |x| \\
 & \text{subject to : } \frac{CV(x)}{CV(M)} = 1
 \end{aligned} \tag{7}$$

Note that the objectives in the optimisation model (7) are contradictory since a lower number of significant cases means a higher error rate and vice versa, that is the greater the number of variables the smaller the error rate. The solution to model (7) is a set of $m \leq X$ non-dominated solutions $C = \{x^k, k \in S\}$, $S = \{1, \dots, X\}$, where each

solution x^k of C represents the best collection of significant k cases. From the practical point of view and in order to simplify the model, it is interesting to sacrifice accuracy slightly when the number of cases are reduced significantly.

3.2 Multiobjective Evolutionary Algorithms

Evolutionary Algorithms (EAs) have been recognised as appropriate techniques for multiobjective optimisation because they perform a search for multiple solutions in parallel [5,7]. Current evolutionary approaches for multiobjective optimisation consist of multiobjective EAs based on the Pareto optimality notion, in which all objectives are optimised simultaneously to find multiple non-dominated solutions in a single run of the EA. These MOEAs usually incorporate diversity mechanisms in order to find non-dominated solutions uniformly distributed on the Pareto front. The decision maker can then choose the most appropriate solution according to the current decision environment at the end of the EA run. Moreover, if the decision environment changes, it is not always necessary to run the EA again. Another solution may be chosen out of the set of non-dominated solutions that has already been obtained.

We propose the use of NSGA-II [6] and SPEA-2 [24], two Multiobjective Evolutionary Algorithms to solve the problem (7). Both NSGA-II and SPEA-2 algorithms have been implemented with the following common components:

- Representation of solutions: a binary codification of fixed length equal to the number of cases in the database is used. In this way, a gene of value 1 in the locus i of the chromosome means that the case x_i has been selected, while 0 means that the case x_i has not been selected.
- Initial population: the initial population is generated randomly using a uniform distribution in the domain.
- Evaluation functions: NSGA-II and SPEA-2 algorithms minimise the following two evaluation functions:

$$\begin{aligned} f_1(x) &= ER(x) \\ f_2(x) &= |x| \end{aligned}$$

where $ER(x)$ is the error ratio obtained using the KNN according to expressions 4 - 6 and $|x|$ is computed as the number of genes with a value 1 of the chromosome x .

- Constraint Handling: both NSGA-II and SPEA-2 algorithms incorporate a constraint handling method described in [6] and [24] respectively. Constraint $g_1(x) = 1$, with $g_1(x) = \frac{CV(x)}{CV(M)}$, is then implemented to ensure that all instances of the output class are covered by solutions.
- Genetic operators: uniform cross and the uniform mutation genetic operators [6] are used in both NSGA-II and SPEA-2 algorithms.
- EA parameters: table 1 shows the values of the EA parameters used in the experiments. These values have been fixed according to the methodology suggested in [13].

Table 1. Parameters of NSGA-II algorithm

Parameters of NSGA-II algorithm	Values	Parameters of NSGA-II algorithm	Values
Size of population	200	Probability of uniform crossover P_c	0.1
Number of generations	500	Probability of uniform mutation P_M	0.6

Table 2. Resume of characteristics of the collected population

Characteristic	Value	Characteristic	Mean	Min	Max	Std.
Male:Female ratio	1.38:1	Age(Years)	46	13	88	19.08
Patients with infection	62.92%	TBSA	38.3	0	85	23.43
Survival Rate	59.50%	SAPS	20.02	0	58	9.59

4 Application to Burn Intensive Care Unit

The Burn Intensive Care Unit (BICU) is responsible for providing medical attention to patients in a critical state. In BICUs, there is a myriad of pathologies with no clear etiology that allows to establish an early diagnosis and therapy [18,21]. Knowledge-intensive techniques, such as rule-based and model-based approaches, imply a costly knowledge acquisition process in terms of time and resources. Others, like statistical strategies require initially high volumes of data, which are not always available. Unlike other AI approaches, CBR provide a simple but effective way to obtain results based on analogy, specially in critical medical scenarios.

4.1 Medical Problem

During the first hour, after the beginning of hypotension and circulatory support, the administration of effective antimicrobial therapy was found to be a critical determinant of survival [18]. Unlike acute myocardial infarction, the presentation of septic shock (severe sepsis and shock) is more unspecific and ambiguous. To detect this problem is even more difficult in patients with severe burn wounds where systemic inflammation response syndrome (SIRS) may also be present. This syndrome is produced by an unknown infection that must be identified. Therefore, the presence of an infection, the skin damages, and the comorbidities are also essential to evaluate a patient. In short, diagnosis of these problems avoiding antibiotic overtreatment requires a long experience and continuous observation [21].

Under medical supervision, a total of 89 patients admitted at the BICU with major burns and complications by infections were selected from the University Hospital of Getafe. Some characteristics of the collected population are shown in table 2.

The clinical experience of the intensive care physicians showed that there are certain facets of clinical evidence, before and after patient admission at the BICU, that seem to be essential for predicting the survival of infected patients. Therefore, data were filtered considering these parameters (see Table 3). In this database, the infection variable indicates the presence of one of the following infections: pneumonia, infection by surgical or burn wounds, or bacteraemia.

Table 3. Parameters that describe patients

Parameters	Description	Parameters	Description
Infection	Presence of an infection	Hepatic-Comorb	Previous liver problems
Total	Total burned surface area (TBSA)	Cardiac-Comorb	Previous heart problems
Prof	Deep burned surface area	Respiratory-Comorb	Previous respiratory problems
Gender	Male/Female	Renal-Comorb	Previous kidney problems
Weight	Patient weight (Kg)	AH	Arterial hypertension
Age	Patient age (years)	Diabetes	Presence of diabetes
Inhibitor	Use of inhibitors	SAPS II	Severity score
HIV-drugs	Drug dependency and HIV		

Due to the reduced volume of this database, statistical approaches will not always provide accurate results for survival assessment. Furthermore, previous works in BICU domain [10] based on intensive knowledge acquisition cannot be included since clinical problems do not always make for etiological consensus. Therefore, the use of CBR techniques seems a suitable approach to solve this problem.

4.2 Experiment Results

In the BICU experiments (Table 4), ENN and RENN have the lowest error, moreover the improvement in sensitivity and specificity is significant. However, they obtain the lowest reduction rate. DROP 1 and 2 reduce more than 75% the case memory with a slight increment of the error rate respect to the original classifier, nevertheless DROP3 has the biggest reduction and significant accuracy results. CNN and RNN achieve relevant error and reduction rates, though they have worse specificity and sensitivity values. The behaviour of MOEA is very similar to the original case memory with less cases, however they need a high run-time for their case selection techniques. In both cases Kappa coefficient has an acceptable value but moderate for the clinical domain.

Table 4. Decision score results of the BICU experiments: err. is the error rate, ρ is the reduction rate, time (in seconds), Sens. is the sensitivity, Spec. is the specificity and κ is the Kappa coefficient

	Err.	ρ	time	Sens.	Spec.	κ
None	0,303	0	0	0,657	0,719	0,376
Rand. 25%	0,319	0,75	0	0,637	0,704	0,341
Rand. 50%	0,306	0,5	0	0,653	0,718	0,369
Rand. 75%	0,3	0,25	0	0,66	0,721	0,381
CNN	0,291	0,53	0	0,694	0,716	0,41
RNN	0,293	0,53	0	0,691	0,714	0,405
ENN	0,212	0,262	0	0,883	0,768	0,601
RENN	0,212	0,262	0	0,883	0,768	0,601
All-KNN	0,283	0,301	0	0,676	0,742	0,418
IB2	0,273	0,673	0	0,706	0,738	0,444
IB3	0,303	0,573	0	0,642	0,732	0,374
Shrink	0,378	0,591	0	0,663	0,714	0,378
DROP1	0,333	0,798	0	0,618	0,692	0,31
DROP2	0,323	0,838	0	0,636	0,697	0,333
DROP3	0,263	0,885	0	0,703	0,758	0,461
SPEA2	0,302	0,516	7,523	0,659	0,719	0,378
NSGA-II	0,303	0,498	5,159	0,656	0,718	0,374

5 Conclusions

In this paper we propose a multiobjective constrained optimisation model for case selection problem. Moreover, we propose the use of evolutionary algorithms to solve this optimisation problem. In order to evaluate the suitability of this approach, we also present an evaluation methodology for case selection algorithms. Classical case selection methods and the evolutionary approach have been evaluated for a particular medical problem in a BICU.

The multiobjective constrained optimisation approach allows to capture a set of non dominated solutions according to accuracy and compactness criteria, and constrained by the coverage assumption. In this way, a decision maker (e.g. physician) can choose, in *a posteriori* decision process, the most appropriate solution according to the current decision environment. For example, a decision maker can sacrifice accuracy slightly when the number of cases are reduced significantly.

Evolutionary algorithms are specially suitable for this type of optimisation problems. Therefore, the well-known multiobjective evolutionary algorithms NSGA-II and SPEA-2 have been adapted in this paper to the particularities of the proposed case selection optimisation problem.

In some works, the evaluation of the case selection process focuses on the validation of the complete system using an arbitrary repetition of the Hold-Out technique [19,20]. Our methodology proposes the use of a KNN classifier, avoiding the use of the complete system. The works described in [14,17] also suggest the same strategy but the case selection process is stopped when the final case memory reaches a determinate size. Unlike our methodology, the case memory size must be known beforehand and this assumption could not be acceptable in some domains. Moreover we suggest the use of a Cross-Validation, providing a more flexible approach by tuning the folder size.

The experiments carried out in the BICU domain show a clear proof of the potential application of the methodology proposed. The results obtained highlights the advantages of the multiobjective evolutionary approach.

Future works will focus on the proposal of novel case selection methods specifically designed for the medical field, and their evaluation with the proposed evaluation methodology.

References

1. Aha, D.W.: Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies* 36, 267–287 (1992)
2. Aha, D.W., Kiblerand, D., Albert, M.K.: Instance-based learning algorithms. *Machine Learning* 6, 37–66 (1991)
3. Ahn, H., Kim, K., Han, I.: A case-based reasoning system with the two-dimensional reduction technique for customer classification. *Expert Systems With Applications* 32, 1011–1019 (2007)
4. Chang, C.L.: Finding prototypes for nearest neighbor classifiers. *IEEE Transactions on Computers* C 23, 1179–1184 (1974)
5. Coello Coello, C.A., Lamont, G.L., van Veldhuizen, D.A.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. In: *Genetic and Evolutionary Computation*, 2nd edn. Springer, Heidelberg (2007)

6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 182–197 (2002)
7. Deb, K., Kalyanmoy, D. (eds.): *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York (2001)
8. Hart, P.E.: The condensed nearest neighbor rule. *IEEE Transaction on Information Theory* 14, 515+ (1968)
9. Jara, A., Martinez, R., Viguera, D., Sanchez, G., Jimenez, F.: Attribute selection by multi-objective evolutionary computation applied to mortality from infection severe burns patients. In: *Proceedings of the International Conference of Health Informatics (HEALTHINF 2011)*, Algarbe, Portugal, pp. 467–471 (2011)
10. Juarez, J.M., Campos, M., Palma, J., Marin, R.: Computing context-dependent temporal diagnosis in complex domains. *Int. J. Expert Sys. with App.* 35(3), 991–1010 (2007)
11. Kolodner, J.L.: Making the Implicit Explicit: Clarifying the Principles of Case-Based Reasoning. In: *Case-based Reasoning: Experiences, Lessons and Future Directions*. ch. 16, pp. 349–370. American Association for Artificial Intelligence (1996)
12. Kuncheva, L.I., Jain, L.C.: Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern Recognition Letters* 20, 1149–1156 (1999)
13. Laumanns, M., Zitzler, E., Thiele, L.: On the Effects of Archiving, Elitism, and Density Based Selection in Evolutionary Multi-objective Optimization. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) *EMO 2001*. LNCS, vol. 1993, pp. 181–196. Springer, Heidelberg (2001)
14. McKenna, E., Smyth, B.: Competence-guided Case-base Editing Techniques. In: Blanzieri, E., Portinale, L. (eds.) *EWCBR 2000*. LNCS (LNAI), vol. 1898, pp. 186–197. Springer, Heidelberg (2000)
15. McSherry, D.: Automating case selection in the construction of a case library. *Knowledge-Based Systems* 13, 133–140 (2000)
16. Nersessian, N.: The Cognitive Basis of Model-based Reasoning in Science. In: *The Cognitive Basis of Science*. ch. 7. Cambridge University Press (2002)
17. Pan, R., Yang, Q., Pan, S.J.: Mining competent case bases for case-based reasoning. *Artificial Intelligence* 171, 1039–1068 (2007)
18. Parrillo, J.E.: Septic shock - vasopressin, norepinephrine, and urgency. *The New England Journal of Medicine* 358(9), 954–956 (2008)
19. Ritter, G.L., Woodruff, H.B., Lowry, S.R., Isenhour, T.L.: Algorithm for a selective nearest neighbor decision rule. *IEEE Transactions on Information Theory* 21, 665–669 (1975)
20. Smyth, B., Keane, M.T.: Remembering to forget - A competence-preserving case deletion policy for case-based reasoning systems. In: *14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*, Montreal, Canada (August 1995)
21. Thombs, B.D., Singh, V.A., Halonen, J., Diallo, A., Milner, S.M.: The effects of preexisting medical comorbidities on mortality and length of hospital stay in acute burn injury: evidence from a national sample of 31,338 adult patients. *Ann. Surg.* 245(4), 626–634 (2007)
22. Tomek, I.: An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems Man and Cybernetics* 6, 448–452 (1976)
23. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. *Machine Learning* 38, 257–286 (2000)
24. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, Athens, Greece, pp. 95–100 (2001)

The VoiceApp System: Speech Technologies to Access the Semantic Web

David Griol, José Manuel Molina, and Víctor Corrales

Universidad Carlos III de Madrid
28911, Leganés, Spain

dgriol@inf.uc3m.es, molina@ia.uc3m.es, 100048294@alumnos.uc3m.es

Abstract. Maximizing accessibility is not always the main objective in the design of web applications, specially if it is concerned with facilitating access for disabled people. In this paper we present the *VoiceApp* multimodal dialog system, which enables to access and browse Internet by means of speech. The system consists of several modules that provide different user experiences on the web. *Voice Dictionary* allows the multimodal access to the Wikipedia encyclopedia, *Voice Pronunciations* has been developed to facilitate the learning of new languages by means of games with words and images, whereas *Voice Browser* provides a fast and effective multimodal interface to the Google web search engine. All the applications in the system can be accessed multimodally using traditional graphic user interfaces such as keyboard and mouse, and/or by means of voice commands. Thus, the results are accessible also for motor-handicapped and visually impaired users and are easier to access by any user in small hand-held devices where graphical interfaces are in some cases difficult to employ.

Keywords: Dialog Systems, Multimodality, VoiceXML, XHTML+Voice, Web Interfaces, Speech Interaction.

1 Introduction

Continuous advances in the development of information technologies and the miniaturization of devices have made it possible to access information and web services from anywhere, at anytime and almost instantaneously through wireless connections. Devices such as PDAs and smartphones are widely used today to access the web, however the contents are accessible only through web browsers, which are operated by means of traditional graphical user interfaces (GUIs). This makes it difficult to use due to the reduced size of the screen and keyboards, and also makes them less usable by motor-handicapped and visually impaired users.

Multimodal interfaces go a step beyond GUIs by adding the possibility to communicate with the devices through other interaction modes such as speech. Multimodal dialog systems [1] can be defined as computer programs designed to emulate communication capabilities of a human being including several communication modalities. The usage of these systems provides three main benefits.

Firstly, they facilitate a more natural human-computer interaction, as it is carried out by means of a conversation in natural language. Secondly, multimodal interfaces make possible the use of these applications in environments in which the use of GUI interfaces is not effective, for example, in-car environments. Finally, these systems facilitate the access to the web for people with visual or motor disabilities, allowing their integration and the elimination of barriers to Internet access [2].

In literature there are two main approaches to develop multimodal dialog systems to access web contents and services. On the one hand, some authors have developed ad-hoc solutions focused on specific tasks, as e-commerce [3], chat functionalities [4], healthcare services [5], surveys [6], or recommendation systems [7]. On the other hand, it is possible to add a speech interface to an existing web browser [8]. This approach may acquire additional complexity in the case of Information Retrieval and Question Answering systems, such as in [9]. However, these works usually emphasize on the search of documents and not on the interaction with the user.

In this paper we describe the *VoiceApp* multimodal dialog system. The system has been developed as a common ground with different web-based applications that can be easily accessed by means of a sophisticated interface which merges voice with traditional GUIs. The idea behind it is to provide an easily extensible common place to create and evaluate multimodal interfaces for web applications. All the applications in *VoiceApp* are easily interoperable so that they are very useful to evaluate the potential of voice interaction in several domains, through a variety of resources and with different users. In the current implementation of the system, the dialog systems have been developed using the XHTML+Voice (X+V) language¹. This language combines the visual modality offered by the XHTML language and the functionalities offered by the VoiceXML language² for the interaction by means of speech. One of the main objectives of the system is to adequately convey to users the logical structure and semantics of content in web documents, and provide them with easy ways to select which parts of a document to listen to.

We will describe the main three applications of *VoiceApp*, although up to five applications have already been implemented. *Voice Dictionary* receives from the user the search criteria and performs a search in the Wikipedia encyclopedia, collects and processes the result of the search, and communicates it to the user by means of visual modalities and synthesized speech. This application also allows to carry out a new search or select any of the links in the result page by using speech or keyboard and mouse. *Voice Browser* is a complete speech-based web search engine. This application collects the topic that the user wants to search on the Internet, communicates this information to the Google search engine, process the resulting information, and communicates it to the user. This application also facilitates multimodal access to the links included in the result

¹ <http://www.w3.org/TR/xhtml+voice/>

² <http://www.w3.org/TR/voicexml20/>

of the search. Finally, *Voice Pronunciations* includes different multimedia games designed for learning foreign languages.

2 Extending Web with Voice

Hyper Text Markup Language (HTML) is the language popularly used for marking up information on the World Wide Web so that it can be displayed using commercially available browsers. However, the World Wide Web Consortium (W3C) realized that HTML was a weak mark-up language if a user wants to process web pages further, given that the use of this language to automatically infer any kind of semantic information requires the analysis of the contents of the web page. This way, the eXtensible Markup Language (XML) was developed as a solution to correct the limitation of the use of information that is available on the web by marking-up information in the web pages and also allowing developers to define their own tags with well-defined meanings that others can understand. The use of an XML-based language significantly improves the portability and interoperability of the programmable parts (including data and programs) of the service.

Many XML-based languages are currently standardized to specify various services (for instance; WSDL for Web Services, ebXML for electric commerce, or CPL for VoIP). VoiceXML is one of these significant standards, as far it makes the Internet accessible through speech, using well-defined semantics that preserves the author's intent regarding the behavior of interactions with the user and facilitating the access to the net in new devices and environments (thus making XML documents universally accessible). VoiceXML audio dialogs feature synthesized speech, digitized audio, recognition of spoken and DTMF key input (Dual-tone multi-frequency signaling), recording of spoken input, telephony, and mixed initiative conversations. The standard also enables the integration of voice services with data services using the client-server paradigm. In addition, many VoiceXML platforms are currently available for research and business use purposes (e.g., Voxeo³).

3 The VoiceApp Multimodal Dialog System

The *VoiceApp* system consists of a set of X+V documents. Some of them are stored from the beginning in the server of the application, while others are dynamically generated using PHP and JavaScript. This dynamic generation takes into account the information extracted from different web servers and MySQL databases in the system, and a set of users preferences and characteristics (e.g., sex, preferred language for the interaction, number of previous interactions with the system, and preferred application). Previous interactions of the users are also taken into account to adapt the system, considering users' most used application, recent topics searched using the application, or errors detected after each interaction with the system.

³ <http://evolution.voxeo.com/>

In order to interact with the X+V documents that make up the system, a web search engine supporting speech interaction and the specifications of this language is required. There are different models for implementing this multimodal interaction on mobile devices. The fat client model employs embedded speech recognition on the specific device and allows conducting speech processing locally. The thin client model involves speech processing on a portal server and is suitable for mobile phones. The implementation of the *VoiceApp* multimodal application for both computers and mobile devices is based on the fat client model, including a multimodal browser and embedded speech recognition on the corresponding device, and a web application server in which the system is stored.

The Opera browser⁴, which allows multimodal web navigation by means of speech, has been integrated for the interaction with the system using a computer. This way, users only need to connect to the application using Opera Voice in a computer with a functioning sound card and loudspeakers or headphones. Opera Voice allows the control of the Opera's interface by talking to the browser. Any ordinary browser command can be done by voice, such as refreshing a web page, navigating to and following the next link in a document, going to the next slide in an Opera Show presentation, or logging on to a password protected Web site. The voice modules that Opera downloads contain two voice types; standard, and high quality. Both of these are able to produce male, female, and child voices.

VoiceApp has also been integrated to facilitate its use by means of mobile phones and hand-held devices. In this case, the system uses the multimodal NetFront Browser v4.1⁵. NetFront supports advanced mobile voice recognition technologies based on X+V, including voice synthesis and voice recognition of mobile Internet data in voice supported web pages. Speech recognition is provided by the embedded ViaVoice speech-recognition program.

3.1 Generation of the XHTML+Voice Pages

The development of oral interfaces implemented by means of X+V implies the definition of grammars, which delimit the speech communication with the system. The `<grammar>` element is used to provide a speech or DTMF grammar that specifies a set of utterances that a user may speak to perform an action or supply information, and for a matching utterance, returns a corresponding semantic interpretation. We have defined a specific strategy to cover the widest range of search criteria in *VoiceApp* by means of the definition of speech recognition grammars in the different applications. This strategy is based on different aspects such as the dynamic generation of the grammars built from the results generated by the interaction with a specific application (e.g., to include the results of the search of a topic using the *Voice Browser*), the definition of grammars that includes complete sentences to support the naturalness of the interaction

⁴ <http://www.opera.com/>

⁵ http://www.access-company.com/products/internet_appliances/netfrontinternet/

with the system (e.g., to facilitate a more natural communication and cover more functionalities in *Voice Pronunciation*), and the use of the ICAO phonetic alphabet⁶ in the cases in which spelling of the words is required in order not to restrict the contents of the search or in situations in which repetitive recognition errors are detected (e.g., in order not to delimit the topics to search using Voice Browser).

Figure 1 shows the translation between a HTML document and its equivalent X+V file. This translation is automatically carried out by means of the PHP files included in *VoiceApp*. As it can be observed, a VoiceXML application consists of one or more scripts that can call each other. A `<form>` is a basic dialog element to present information and gather user inputs, which is generally composed of several form items. The form items are subdivided into input items and control items. Variables in VoiceXML are declared by `<var>` elements, or by form items such like `<field>` with name attributes. VoiceXML has several elements to operate the control flow of the script (for example, `<if>`, `<goto>`, `<exit>`, and `<submit>`). Event handling is carried out by means of elements like `<noinput>` and `<nomatch>`.

3.2 Voice Dictionary, Voice Browser and Voice Pronunciation

As previously described, the *Voice Dictionary* application offers a single environment where users can search contents in the Wikipedia encyclopedia with the main feature that the access to the application and the results provided by the search are entirely facilitated to the user either through visual modalities or by means of speech. Once the result of an initial search is displayed on the screen and communicated to the user by means of speech, they can easily access any of the links included in the result of the search or visit the rest of applications in the system with the possibility of interrupting the system's speech in any case. This functionality is achieved by means of the dynamic generation of the corresponding grammars, in which the different links that are present in the result of a specific search are included in the dynamic X+V page automatically generated by means of a PHP script that captures the different information sources to inform the user about them (headings, text, contents, formulas, links, etc.). Figure 2 shows the initial page of the application.

Google is currently one of the most important companies for the management of information on the Internet due to its web search engine and a number of applications and services developed to access information on the net. This way, the *Voice Browser* application has been developed with the main objective of allowing the speech access to facilitate both the search and presentation of the results in the interaction with the Google search engine. The application interface receives the contents provided by the user and displays the results both visually and using synthesized speech. The application also allows the multimodal selection of any of the links included in the result of the search by numbering them and allowing using their titles as voice commands (Figure 2).

⁶ International Civil Aviation Organization (ICAO) phonetic alphabet:
<http://www.icao.int/icao/en/trivia/alphabet.htm>

<pre> % HTML document <html> <head> <title>VoiceApp-Voice Browser</title> </head> <body> LINK 1: The Beatles Find out all about The Beatles... ... LINK 10: Songs, Pictures, and Stories of The Beatles Beatles website for collectors and fans ... </body> </html> </pre>	<pre> % XHTML+Voice file <?xml version="1.0" encoding="ISO-8859-1"?> <html xmlns="http://www.w3.org/1999/xhtml" xmlns:vxml="http://www.w3.org/2001/vxml" xmlns:ev="http://www.w3.org/2001/xml-events" xmlns:xv="http://www.voicexml.org/2002/xhtml+voice"> <head> <title>VoiceApp - Voice Browser</title> <vxml:form id="nav"> <vxml:block> To visit the links, you have to say "LINK" and thecorresponding number. </vxml:block> <vxml:field xv:id="app" name="app"> <vxml:grammar src="inig.jsgf"/> <vxml:nomatch> <vxml:prompt> Please repeat again, I can not understand you. </vxml:prompt> </vxml:nomatch> </vxml:field> <vxml:filled mode="all"> <vxml:prompt> Ok got them. </vxml:prompt> <vxml:elseif cond="app == 'home'"/> <assign name="window.location" expr="index"/> <vxml:elseif cond="app == 'link 1'"/> <assign name="window.location" expr="x1x"/> ... <vxml:elseif cond="app == 'link 10'"/> <assign name="window.location" expr="x10x"/> </vxml:if> </vxml:filled> </vxml:form> <script src="java.js" type="text/javascript"></script> </head> <body id="docBody" ev:event="load" ev:handler="#nav"> <div id="cont" ev:event="click" ev:handler="#nav"> <h1>Results for: The Beatles</h1> LINK 1: The Beatles Find out all about The Beatles... ... LINK 10: Songs, Pictures, and Stories of The Beatles Beatles website for collectors and fans ... </body></html> </pre>
---	---

Fig. 1. Translation of a HTML document into an equivalent XHTML+Voice file

The *Voice Pronunciation* application has been developed with the main objective of implementing a web environment that facilitates second-language learning with two games that help to acquire new vocabulary and train the words pronunciation. The game *Words* shows on the screen and synthesizes orally the definition of one of the over one hundred thousand words stored in a database of

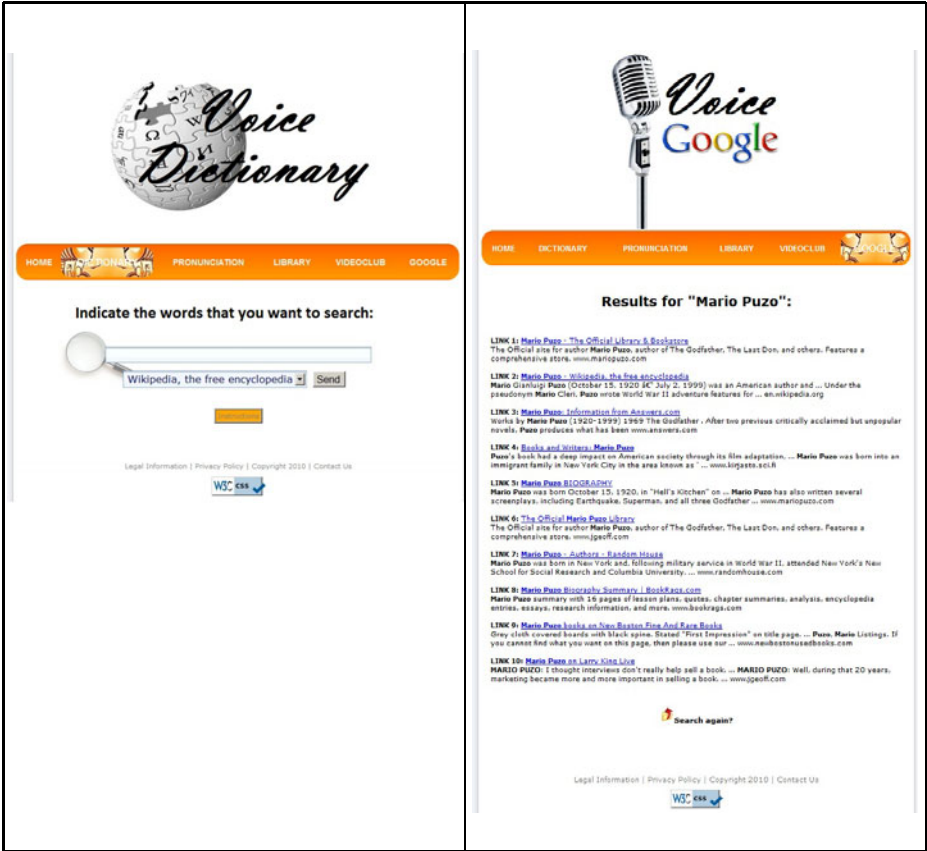


Fig. 2. Main page of the *Voice Dictionary* application and screen showing the result of a search using the *Voice Google* application

the application and the user must guess the word. The game *Pictures* uses images stored in a database and annotated with different difficulties, whose exact name must be correctly uttered by the user to continue in the game and increase the score (Figure 3). The specific problems and errors detected during the previous interactions of the users with this application are taken into account for the selection of the different words and images and to consequently adapt both games to the specific evolution of each user during the learning process.

4 Preliminary Evaluation

A number of tests and verifications have been carried out to maximize the functionalities and accessibility of the different applications included in the *VoiceApp* system. These tests have been very important to detect and correct programming errors and accessibility problems. One of the main identified problems was

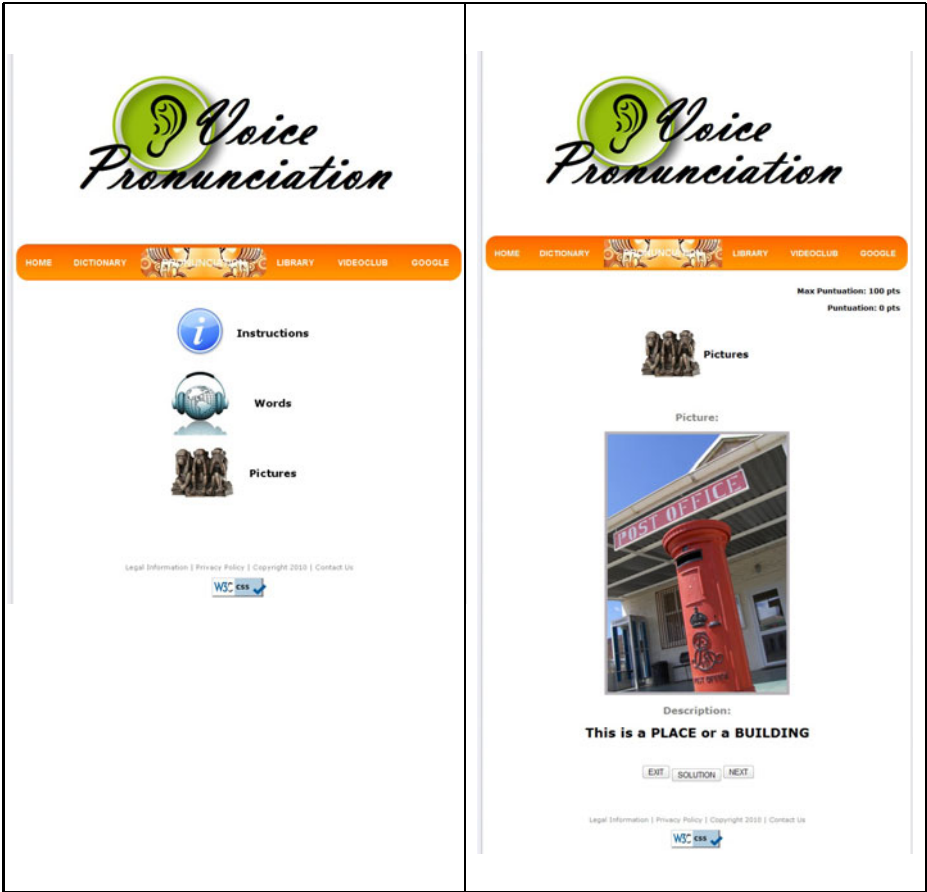


Fig. 3. Main page of the *Voice Pronunciation* application and its Pictures functionality

related to the generation of inconsistencies when words with similar pronunciation were reserved to both interact with by the Opera search engine and the different applications in the system. These inconsistencies have been limited to the maximum so that the possible matches between selected words have been eliminated in the different applications.

In addition, we have completed a preliminary assessment by means of a questionnaire to measure users subjective opinion about the system. The questionnaire contained five questions: i) Q1: *Did the system correctly understand you during the interaction?*; ii) Q2: *Did you understand correctly the messages of the system?*; iii) Q3: *Was it simple to obtain the requested information? / Was it simple to play the game?*; iv) Q4: *Do you think that the interaction rate was adequate?*; v) Q5: *Was it easy to correct mistakes made by the system?*; vi) Q6: *In general terms, are you satisfied with the performance of the system?* The possible answers to the complete set questions were the same: *Never, Rarely, Sometimes*

Usually and *Always*. A numerical value between one and five was assigned for each answer (in the same order as they are shown in the questionnaire). Table 1 shows the average, maximum and minimum values obtained from the results provided by a total of 35 students and professors of our University using the different modules of the system without predefined scenarios.

Table 1. Results of the preliminary evaluation of the *VoiceApp* system (1=minimal value, 5=maximum value)

	Q1	Q2	Q3	Q4	Q5	Q6
Average value	3.6	3.8	3.2	3.7	3.2	4.3
Maximum value	4	5	5	4	4	5
Minimal value	2	3	2	3	2	3

The results of the preliminary evaluation of the *VoiceApp* system show that the users who participated in the assessment positively evaluate the facility of obtaining the requested information by interacting with the system, the appropriate interaction rate during the dialog, and overall operation of the different applications in the system. The main problems mentioned by the users include the need of improving the word error rate and achieve a better clarification of the action expected by the system at each moment of interaction. In addition, the 97% of the interactions finished achieving the objective(s) expected by the user, only the 4% of the systems turns corresponded to reprompts and the 12% to system confirmations. The error correction rate (computer as the average number of corrected errors per dialog divided by the number of corrected and uncorrected errors) was 91%.

5 Conclusions

The *VoiceApp* system has been developed as a framework for the study of the XHTML+Voice technology to develop multimodal dialog systems that improve the accessibility to information on the Internet. The programming languages XML, XHTML and VoiceXML respectively deal with the visual design of the application and allow spoken dialog with the user. This way, multimodal interaction capabilities have been integrated for both the input and output of the system. The use of additional programming languages, as PHP and JavaScript, as well as relational database management systems such as MySQL, facilitates the incorporation of adaptive features and the dynamic generation of contents for the application. Accessibility has been defined as one of the most important design requisites of the system. This way, detailed instructions, help messages and menus have been also incorporated to facilitate the interaction with the different applications in the system.

The set of applications described in this paper respectively facilitate the multimodal access for the search of contents in the Wikipedia encyclopedia, the learning of new languages by improving the words pronunciation by means of

funny games, and the complete implementation of a speech-based interface to an Internet search engine.

Current research lines include the adaptation of the system for its interaction using additional languages, a more detailed assessment of each specific application, and the incorporation of new features in each one of them. Another important research line consists of the adaptation of the different applications taking into account specific user profiles considering more detailed information about their preferences and evolution.

Acknowledgements. Research funded by projects CICYT TIN 2008-06742-C02-02/TSI, CICYT TEC 2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485), and DPS 2008-07029-C02-02.

References

1. López-Cózar, R., Araki, M.: Spoken, Multilingual and Multimodal Dialogue Systems. John Wiley & Sons (2005)
2. Beskow, J., Edlund, J., Granström, B., Gustafson, J., Skantze, G., Tobiasson, H.: The MonAMI Reminder: a spoken dialogue system for face-to-face interaction. In: Proc. of Interspeech/ICSLP, pp. 296–299 (2009)
3. Tsai, M.: The VoiceXML dialog system for the e-commerce ordering service. In: Shen, W.-m., Chao, K.-M., Lin, Z., Barthès, J.-P.A., James, A. (eds.) CSCWD 2005. LNCS, vol. 3865, pp. 95–100. Springer, Heidelberg (2006)
4. Kearns, M., Isbell, C., Singh, S., Litman, D., Howe, J.: CobotDS: A Spoken Dialogue System for Chat. In: Proc. of AAAI 2002, pp. 425–430 (2002)
5. Griol, D., McTear, M.F., Callejas, Z., López-Cózar, R., Ábalos, N., Espejo, G.: A Methodology for Learning Optimal Dialog Strategies. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 507–514. Springer, Heidelberg (2010)
6. Stent, A., Stenchikova, S., Marge, M.: Reinforcement learning of dialogue strategies with hierarchical abstract machines. In: Proc. of SLT 2006, pp. 210–213 (2006)
7. Chai, J., Horvath, V., Nicolov, N., Stys, M., Kambhatla, N., Zadrozny, W., Melville, P.: Natural language assistant: A dialog system for online product recommendation. AI Magazine 23, 63–75 (2002)
8. Vesnicer, B., Zibert, J., Dobrisek, S., Pavesic, N., Mihelic, F.: A voice-driven web browser for blind people. In: Proc. of Interspeech/ICSLP, pp. 1301–1304 (2003)
9. Mishra, T., Bangalore, S.: Qme!: a speech-based question-answering system on mobile devices. In: Proc. of HLT 2010, pp. 55–63 (2010)

A Cluster Based Pseudo Feedback Technique Which Exploits Good and Bad Clusters

Javier Parapar and Álvaro Barreiro

IRLab, Computer Science Department
University of A Coruña, Spain
{javierparapar,barreiro}@udc.es

Abstract. In the last years, cluster based retrieval has been demonstrated as an effective tool for both interactive retrieval and pseudo relevance feedback techniques. In this paper we propose a new cluster based retrieval function which uses the best and worst clusters of a document in the cluster ranking, to improve the retrieval effectiveness. The evaluation shows improvements in some standard TREC collections over the state-of-the-art techniques in precision and robustness.

1 Introduction and Motivation

Several strategies were studied in the history of the Information Retrieval in order to improve the retrieval models effectiveness. One technique that has been demonstrated successful is relevance feedback. In this family of techniques is particularly interesting the so called pseudo relevance feedback [3], where relevance of the feedback documents is assumed.

Clustering has been considered as an useful tool in the retrieval process since the formulation of the cluster hypothesis in 1979 [16]. This hypothesis states that very related documents tend to be relevant to the same query. Indeed, several experiments [5][18] have demonstrated that clustering algorithms working at pseudofeedback time can obtain clusters with a high percentage of relevant documents, still the automatic identification of these clusters between the whole set of them is still a challenge.

Although initial experiments using query specific clustering [12] in order to improve the retrieval effectiveness were not conclusive, after improving the cluster representation [13] and with the use of clustering algorithms that support overlapping [9], finally the quality of the initial ranking was significant improved with cluster based re-ranking [13][7].

It was only recently when a cluster based retrieval approach was used to improve the quality of the pseudo relevance set, for further use in query expansion methods [11]. This approach takes advantage of the better initial ranking produced by the cluster based retrieval to select a better pseudo relevance set, improving in this way the effectiveness, sometimes significantly. But, although this kind of methods tend to improve the effectiveness in average, one known problem of them is the lack of robustness, i.e., still a significant amount of queries

are negatively affected by them. One of the main factors of this behaviour is the presence of non-relevant documents in the feedback set.

In this paper we present a new cluster based retrieval method that exploits bad clusters in order to reduce the amount of non-relevant documents in the feedback set. We consider not just if a document is present inside a “good” cluster to update its score, but also the presence of the document in “bad” (low relevance score) clusters. As far as we know this kind of negative information has not been exploited yet in the context of pseudo relevance feedback.

We tested our approach in several TREC Collections and compared with a Language Modelling retrieval approach [20], a query expansion based model [1] and with the resampling method presented in [11]. The evaluation shows that the results in terms of MAP are so good or better than the resampling approach but our method consistently improves the robustness in all the collections.

The paper is presented as follows. Section 2 presents our proposal explaining the different steps of the model. Section 3 explains the evaluation methodology and comments the results. In Section 4 we describe the related work and finally conclusions and future work are reported in Section 5.

2 Cluster Based Pseudo Relevance Feedback

In order to get a better pseudo relevance set we formulated a new cluster based re-ranking function. The first step of our method is to perform an initial document ranking, in this case we chose as base a Language Model (LM). After that, we cluster the top N documents (d_{init}), we chose in this case a clustering algorithm with overlapping. Once the top documents are clustered the query likelihood is calculated for the resulting clusters. After that, the clusters query likelihoods are combined by the retrieval formula re-ranking the documents. And finally these new top documents are used to feed a query expansion process.

Initial Ranking. In Language Models, the probability of a document given a query, $P(d|q)$, is estimated using Bayes’ rule as presented in Eq. 1.

$$P(d|q) = \frac{P(q|d) \cdot P(d)}{P(q)} \stackrel{rank}{=} \log P(q|d) + \log P(d) \quad (1)$$

In practice $P(q)$ can be dropped for document ranking purposes. The prior $P(d)$ encodes a-priori information on documents and the query likelihood, $P(q|d)$, incorporates some form of smoothing. In this paper we consider uniform priors and unigram Language Models with Dirichlet smoothing [20], see Eq. 2.

$$P(q|d) = \prod_{i=1}^n \frac{tf(q_i, d) + \mu \cdot P(q_i|Col)}{|d| + \mu} \quad (2)$$

where n is the number of query terms, $tf(q_i, d)$ is the raw term frequency of q_i in d , $|d|$ is the document length expressed in number of terms, and μ is a parameter for adjusting the amount of smoothing applied. $P(q_i|Col)$ is the probability of the term q_i occurring in the collection Col that we obtained with the maximum likelihood estimator computed using the collection of documents.

Clustering Algorithm. Once that the initial ranking is obtained, clustering is performed over the top N documents. The use of clustering algorithms with overlapping has already been demonstrated successful [9] in cluster based retrieval. Indeed, initial approaches to query specific cluster [12] were not conclusive and it was only after incorporating clustering algorithms with overlapping [13] when the results were improved. As we explained one of the main points of our method is to use the information provided by bad clusters to avoid non-relevant documents in the pseudo relevance set. In order to do this we used a clustering algorithm that supports overlapping, i.e. one document can belong to one or more clusters.

The straightforward selection based in previous works could be using a k -nearest neighbours (k -NN) algorithm, but the k -NN forces to each document to have k neighbours. This aspect is not desired in our approach because we will exploit that a document belongs to a low scored cluster. If we had used k -NN, a non-relevant document with low query likelihood and no close neighbours could attract other documents that, although they are not close to that document, they are the closest ones.

So we decided to cluster the documents in base to a given threshold t , grouping for each document those neighbours that are more similar than t . Let's call this way of grouping thr -N. The purpose of this algorithm is that non-relevant documents could be isolated in singletons [14]. Standard $tf \cdot idf$ document representation was used with the well-known cosine similarity function in order to calculate distances between the documents.

Cluster Query Likelihood. In order to exploit the cluster information in our retrieval approach we need a way of estimating cluster the query likelihood. In the origin the first approaches to cluster retrieval considered the clusters as meta-documents, i.e. one cluster is represented as the concatenation of the documents that belong to it [9,12], or the centroid of the cluster [19]. But these representations suffer from several problems because of the document and cluster sizes. As demonstrated by Liu and Croft in [13], the geometric mean is a better cluster representation in order to calculate the cluster query likelihood, so it was chosen in our approach. The cluster query likelihood based on the geometric mean representation was calculated combining equations 3 and 4.

$$P(q|C) = \prod_{i=1}^n P(q_i|C) \tag{3}$$

$$P(w|C) = \prod_{i=1}^{|C|} P(w|d_i)^{\frac{1}{|C|}} \tag{4}$$

where n is the number of query terms, $|C|$ is the number of documents in the cluster, and $P(w|d_i)$ was computed using a Dirichlet estimate. Finally the cluster query likelihood applying logarithmic identities can be calculated as in Eq. 5

$$P(q|C) = \prod_{i=1}^n e^{\frac{\sum_{i=1}^{|C|} \log P(w|d_i)}{|C|}} \tag{5}$$

Cluster Based Reranking. Previous approaches to cluster based re-ranking only used the presence of a document in a good cluster as indicator of its relevance. As previous explained these approaches when using to construct pseudo relevance sets for further processing with, for instance query expansion, suffer from the problem that even the good clusters are not one hundred percent composed of relevant documents. The inclusion of non-relevant documents in the relevance set will produce a poor performance of the query expansion process resulting in effectiveness degradation for that query.

The final objective of our approach is to reduce the number of non-relevant documents in the pseudo relevance set. To achieve that point we decided to use the information given by the bad clusters. Our hypothesis is that given two documents d_1 and d_2 , and being C_{1max} , C_{1min} , C_{2max} and C_{2min} the clusters with best and worst query likelihood to which d_1 and d_2 belong respectively, if $P(q|C_{1max}) = P(q|C_{2max})$ and $P(q|d_1) = P(q|d_2)$ then if $P(q|C_{1min}) > P(q|C_{2min})$ should indicate that d_1 is likely to be more relevant than d_2 . In other words if a document belongs to low clusters in the cluster ranking, it should be a pseudo negative indicator about its relevance.

So in order to produce a document ranking we decided to combine the document query likelihood, with the pseudo positive information in terms of best cluster, and the negative in terms of the worst cluster to which the document belongs. The query likelihood combination is presented in Eq. 6

$$P'(q|d) = P(q|d) \times \max_{d \in C_i} P(q|C_i) \times \min_{d \in C_i} P(q|C_i) \tag{6}$$

where $P(q|d)$ was estimated as in Eq. 2 and $P(q|C_i)$ was estimated as in Eq. 5

Ideally removing all the non-relevant documents from the relevant set would have a great impact in order to get better expanded queries and, as a consequence, to improve the final retrieval effectiveness. Even although some relevant documents could be penalised because they group with other ones which appear low in the ranking, this effect will be extensively compensated by the benefit of removing the non-relevant documents from the relevance set.

Query Expansion for Pseudo Relevance Feedback. Once that we obtained a ranking with, hopefully, less amount of non-relevant documents in high positions, we take the first $|r|$ documents as the pseudo relevance set. With this relevance set we feed a query expansion approach. We chose to use Kullback Leiber Divergence (KLD) as the scoring function to select expansion terms [115], so the e terms with highest KLD score, calculated as in Eq 7, are selected to expand the original query.

$$kld_{score}(t) = \frac{tf(t, r)}{NT_r} \times \log \frac{tf(t, r) \times |Col|}{NT_r \times tf(t, Col)} \tag{7}$$

where $tf(t, r)$ is the term frequency of t in the pseudo relevance set, NT_r is the number of terms in the pseudo relevance set r , $|Col|$ is the total number of terms in the collection and $tf(t, Col)$ is the term frequency of t in the whole collection.

We interpolated the e terms selected as results of the KLD scoring formula for expansion terms with the original query. That was already demonstrated successful in RM3 [4], obtaining a high performance query expansion model. Therefore, the final rank is processed with the expanded query presented in [8]

$$\lambda \times (q_1, \dots, q_n), (1 - \lambda) \times (kld_{score}(t_1)t_1, \dots, kld_{score}(t_e)t_e) \quad (8)$$

where q_i are the original query terms, $kld_{score}(t_i)t_i$ are the expanded terms with the weight corresponding to their KLD score and λ is a parameter $0 \leq \lambda \leq 1$ than combines the original query and the expanded one.

3 Experiments and Results

The evaluation of our approach was performed over four TREC collections comparing with a baseline retrieval model, a baseline feedback model and a baseline cluster based feedback model. The results of an upper-bound model are also reported.

3.1 Settings and Methodology

Collections. We tested our method (BWCluster_PF) in four collections, two text collections: AP and WSJ, and two web collections: WT10g and GOV. We decided to use cross-validation evaluation: we performed training in a set of topics in the AP for both text collections and individual training for the each web collections. We chose to use short queries (title only). All the collections were preprocessed with standard stopword removal and Porter stemmer. In Table [1] are summarised the evaluation settings.

Table 1. Collections and topics for training and test

Col.	# of Docs	Topics	
		Train	Test
AP	242,918	51-150	151-200
WSJ	173,252		151-200
WT10g	1,692,096	451-500	501-550
GOV	1,247,753	WT04.1-50	WT04.151-200

Baselines

- **LM:** The baseline LM retrieval model with Dirichlet smoothing that was used as base of the other methods.
- **KLQE:** a pseudo relevance model based on the query expansion [1]. The selection of expansion terms and the construction of the expanded query which was explained in section [2]. This was also the query expansion approach used for the next approaches.

- **Resampling:** The cluster based resampling method presented by Lee *et al.* in [11], but using the Geometric Mean instead of the document concatenation to compute the cluster query likelihood and KLQE instead of the estimation of the Relevance Model [10] to compute the expanded query.
- **TrueRF:** An upper-bound of all the pseudo relevance methods, was computed feeding the KLQE approach with all the relevant documents present in the d_{init} .

Training and Evaluation. As commented we performed cross-validation strategy, more precisely we perform training for the text collection with a set of topics with AP dataset, and testing in the AP and WSJ with different topics. For the web collections we trained in each collection with 50 queries each, and testing with other sets of queries. Training was performed optimising Mean Average Precision (MAP).

There are several parameters to train. Namely, the smoothing parameter μ was tuned in the baseline LM retrieval model ($\mu \in \{10, 100, 1000, 2000, 3000, 4000, 5000, 6000\}$) and its value was used for every other retrieval model. The parameters $|r|$, the size of the pseudo relevance set, e , the number of expansion terms, and λ , the interpolation factor, for the pseudo feedback based query expansion were trained in the KLQE model ($|r| \in \{5, 10, 25, 50, 75, 100\}$, $e \in \{5, 10, 25, 50, 75, 100\}$ and $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$) and also used in the cluster based approaches. The cluster based retrieval models, resampling and our method, need apart of mu , $|r|$, e and λ , the parameters k that was set to 5, t that was set to 0.05 and N , the size of the d_{init} , that was set to 100. In [11] Lee *et al.* demonstrated that with exhaustive tuning the resampling method obtains significant improvements over pseudo feedback based query expansion methods. In this paper we decided to avoid excessive tuning effort and we fix the values of mu , $|r|$, e and λ to the one trained in LM and KLQE respectively.

Therefore we have to remark that both cluster based approaches can be improved in terms of effectiveness by specifically tuning every parameter in each of the retrieval approaches. Also, our evaluation methodology allows see more clearly the effect of the cluster information without depending on excessive tuning effort. For the TrueRF upper-bound model we maintained the same parameter set as in the KLQE model only changing r by the set of relevant documents in the d_{init} .

Finally test values are reported for MAP and RI. The Robustness Index (RI) ($-1 \leq RI(q) \leq 1$) also called Reliability of Improvement Index of one model respect a baseline was presented by Sakail *et al.* in [17] and it is formulated as in Eq 9:

$$RI(q) = \frac{n_+ - n_-}{|q|} \quad (9)$$

where q is the set of queries over the RI has to be calculated, n_+ is the number of improved queries, n_- the number of degraded queries and $|q|$ the total number of queries in q .

3.2 Results

Analysing the MAP values for the test topics (see Table 2) it has to be notice that our approach significantly outperforms the baseline LM for every collection, a fact that neither the KLQE nor the Resampling method achieve. Our method only achieves statistically significant improvements over the query expansion method in the WT10g collection, this is explained in part because the KLQE values were obtained with the best parameter settings meanwhile the values of our method did not receive individual parameter tuning. The values of resampling method do not achieve statistical significant improvements over the query expansion method but again the same reason as previously explained applies; we have to remark that such achievements are reported in [11]. In WSJ, WT10g and GOV collection our method outperforms the resampling method being the improvements significant in two of them. Of course the upper-bound model outperforms any other method.

Table 2. Values for Mean Average Precision (MAP) on the test topics. Statistical significant improvements (Wilcoxon $p < 0.05$) with respect to the Dirichlet smoothed LM, KLQE, Resampling and Our method are superscripted with *a,b,c* and *d* respectively.

<i>Col.</i>	MAP				
	<i>LM</i>	<i>KLQE</i>	<i>Resampling</i>	<i>BWCluster_PF</i>	<i>TrueRF</i>
AP	.2078	.2918 ^a	.2889 ^a	.2884 ^a	.4569 ^{abcd}
WSJ	.3366	.3940 ^a	.3927 ^a	.3982 ^{ac}	.5614 ^{abcd}
WT10g	.2101	.2166	.2173	.2224 ^{abc}	.3002 ^{abcd}
GOV	.1949	.2198 ^a	.1994	.2128 ^a	.2200 ^{abcd}

Query robustness values measured with RI over the LM baseline model are reported in Table 3. The first fact to mention is that our method is better or equal than the KLQE approach except in the AP collection, where MAP values were also worse. In the case of the resampling approach the RI values are worse than the KLQE even in the case of the WT10g collection, where the resampling method MAP is better than KLQE. In this case, in the WT10g collection, the RI is negative, this means than more queries are penalised than benefited. Comparing both cluster based methods we have to remark that our method outperforms the resampling method in every collection but the AP where both methods report the same values. Again as expected the query robustness

Table 3. Values for Robustness Index (RI) with respect to the LM baseline model for every collection on the test topics

<i>Col.</i>	RI			
	<i>KLQE</i>	<i>Resampling</i>	<i>BWCluster_PF</i>	<i>TrueRF</i>
AP	0.44	0.40	0.40	0.96
WSJ	0.36	0.28	0.44	0.92
WT10g	0.16	-0.08	0.16	0.80
GOV	0.56	0.44	0.60	1.00

of the TrueRF upper-bound model is greater than any other method, although in three of the collections TrueRF still damage some queries.

4 Related Work

Since the formulation of the cluster hypothesis [16] several works tried to exploit clustering information to improve information retrieval tasks. It was only recently when conclusive results were presented improving retrieval effectiveness using query specific clustering. We have to cite the work of Kurland and Domshlak [8] where several features were aggregated to obtain better high precision in the re-ranking of top documents. Kurland and Domshlak used several features related with cluster information, namely *query faithfulness*, *self faithfulness*, *initial list faithfulness* and *peer faithfulness*. The individual feature results seem indicate that *peer faithfulness* is the better indicator, although the aggregation of all the features reports the best values. An approach based on similar facts is presented by Kurland in [7], in this paper the author present several approaches (aspect models and interpolation models) that also combine information about peer clusters. High precision is again improved, although the performance is quite dependent of the settings and MAP values are also reported but only in a cut-off of 50 documents. In [6] Kurland presented several cluster based re-ranking approaches, exploiting in this case clusters with high percentage of relevant documents. several features are combined resulting in improvements in high precision over the initial ranking.

Recently Lee *et al.* [11] proposed query specific clustering in order to improve the quality of the pseudo relevance set used in the query expansion process, in this case a relevance model (RM) [10]. This method uses as cluster re-ranking method the original cluster query likelihood presented by Liu and Croft in [12] but with overlapping clusters. The results show significant improvements over the initial LM based rank and the RM rank in several collections. This approach that we used in order to compare our method still shows some problems with query robustness that we tried to solve with our alternative approach using pseudo negative information. Although the results reported in [11] are higher than the reported here for their method we have again to remark that we did not perform individual parameter tuning for each retrieval method, and also we used KLD based query expansion instead of RM, so the results on this paper can be still improved for both, resampling and our method.

In line with the need of consider negative information, in this case associated with clustering processes, we have to remark the analysis already presented in 1995 by Lu *et al.* in [14]. In this paper it is commented that after running a cluster algorithm over the the top documents of a rank, most of the singletons (clusters with only one document) are non-relevant documents, and should be removed. This data suggested us that the clustering algorithm should allow the creation of singletons. Really not every singleton contains a non-relevant document but, allowing the creation of singletons, the real non-relevant documents will not be promoted in the ranking benefited because they are clustered with relevant ones, while non affecting negatively when they are relevant documents.

Also recently several works approached the task of getting a better pseudo relevance set, in this case to increase the diversity, but none of them show conclusive results. In [2] Collins-Thompson and Callan present sampling over the top documents based on query variants. The objective of having less redundant pseudo relevance set is also approached in [17]. Sakai *et al.* presented in this case a resampling method that it is actually based on clustering, the top documents are clustered based on the common query terms selecting only some of each cluster in order to improve diversity in the relevance set. But again the results presented in the evaluation are not conclusive.

5 Conclusions and Future Work

The proposed method introduces the use of bad clusters in order to achieve pseudo feedback sets with less non-relevant documents. The pseudo negative information is obtained from the belonging of the documents to a “bad” cluster in a cluster re-ranking approach. The results show improvements in MAP over the existing cluster based approaches for pseudo relevance feedback, that in some settings are statistical significant. Another good result is the improvement in terms of query robustness: our approaches penalise less queries than previous ones.

Further analysis of the non-relevant documents that still remain in the pseudo relevance set has to be done. We also want to study the effect of taking a bigger set of top documents d_{init} (evaluation was done with the top 100 documents) that should be an important factor when considering the pseudo negative information. Also we will perform individual parameter tuning for the cluster based approaches in order to report their best values.

Acknowledgements. This work was funded by *Ministerio de Ciencia e Innovación* under project TIN2008-06566-C04-04.

References

1. Carpineto, C., de Mori, R., Romano, G., Bigi, B.: An information-theoretic approach to automatic query expansion. *ACM Trans. Inf. Syst.* 19(1), 1–27 (2001)
2. Collins-Thompson, K., Callan, J.: Estimation and use of uncertainty in pseudo-relevance feedback. In: *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 303–310. ACM Press, New York (2007)
3. Croft, W., Harper, D.: Using probabilistic models of document retrieval without relevance information. *Journal of Documentation* 35, 285–295 (1979)
4. Diaz, F., Metzler, D.: Improving the estimation of relevance models using large external corpora. In: *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 154–161. ACM, New York (2006)
5. Hearst, M.A., Pedersen, J.O.: Reexamining the cluster hypothesis: scatter/gather on retrieval results. In: *SIGIR 1996: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 76–84. ACM, New York (1996)

6. Kurland, O.: The opposite of smoothing: a language model approach to ranking query-specific document clusters. In: SIGIR 2008: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 171–178. ACM, New York (2008)
7. Kurland, O.: Re-ranking search results using language models of query-specific clusters. *Inf. Retr.* 12(4), 437–460 (2009)
8. Kurland, O., Domshlak, C.: A rank-aggregation approach to searching for optimal query-specific clusters. In: SIGIR 2008: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 547–554. ACM, New York (2008)
9. Kurland, O., Lee, L.: Corpus structure, language models, and ad hoc information retrieval. In: SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 194–201. ACM, New York (2004)
10. Lavrenko, V., Croft, W.B.: Relevance based language models. In: SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 120–127. ACM, New York (2001)
11. Lee, K.S., Croft, W.B., Allan, J.: A cluster-based resampling method for pseudo-relevance feedback. In: SIGIR 2008: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 235–242. ACM, New York (2008)
12. Liu, X., Croft, W.B.: Cluster-based retrieval using language models. In: SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 186–193. ACM, New York (2004)
13. Liu, X., Croft, W.B.: Evaluating Text Representations for Retrieval of the Best Group of Documents. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 454–462. Springer, Heidelberg (2008)
14. Lu, X.A., Ayoub, M., Dong, J.: Ad Hoc Experiments using Eureka. In: Proceedings of the Fifth Text Retrieval Conference (TREC-5), pp. 229–240 (1996)
15. Parapar, J., Barreiro, A.: Promoting Divergent Terms in the Estimation of Relevance Models. In: Amati, G., Crestani, F. (eds.) ICTIR 2011. LNCS, vol. 6931, pp. 77–88. Springer, Heidelberg (2011)
16. Rijsbergen, C.V.: *Information Retrieval*. Butterworths, London (1979)
17. Sakai, T., Manabe, T., Koyama, M.: Flexible pseudo-relevance feedback via selective sampling. *ACM Transactions on Asian Language Information Processing (TALIP)* 4(2), 111–135 (2005)
18. Tombros, A., Villa, R., Van Rijsbergen, C.J.: The effectiveness of query-specific hierarchic clustering in information retrieval. *Inf. Process. Manage.* 38(4), 559–582 (2002)
19. Voorhees, E.M.: The cluster hypothesis revisited. In: SIGIR 1985: Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 188–196. ACM, New York (1985)
20. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.* 22(2), 179–214 (2004)

SAHN with SEP/COP and SPADE, to Build a General Web Navigation Adaptation System Using Server Log Information

Olatz Arbelaitz, Ibai Gurrutxaga, Aizea Lojo,
Javier Muguerza, and Iñigo Perona

Dept. of Computer Architecture and Technology
University of the Basque Country
M. Lardizabal, 1, 20018 Donostia, Spain
{olatz.arbelaitz,i.gurrutxaga,alajo002@ikasle.,
j.muguerza,inigo.perona}@ehu.es

Abstract. During the last decades, the information on the web has increased drastically but larger quantities of data do not provide added value for web visitors; there is a need of easier access to the required information and adaptation to their preferences or needs. The use of machine learning techniques to build user models allows to take into account their real preferences. We present in this work the design of a complete system, based on the collaborative filtering approach, to identify interesting links for the users while they are navigating and to make the access to those links easier. Starting from web navigation logs and adding a generalization procedure to the preprocessing step, we use agglomerative hierarchical clustering (SAHN) combined with SEP/COP, a novel methodology to obtain the best partition from a hierarchy, to group users with similar navigation behavior or interests. We then use SPADE as sequential pattern discovery technique to obtain the most probable transactions for the users belonging to each group and then be able to adapt the navigation of future users according to those profiles. The experiments show that the designed system performs efficiently in a web-accessible database and is even able to tackle the cold start or 0-day problem.

1 Introduction

During the last decades, the information on the web has increased drastically and this often makes the amount of information intractable for users. As a consequence, the need for web sites to be useful in an efficient way for users has become specially important. Larger quantities of data do not provide added value for web visitors; there is a need of easier access to the required information and adaptation to their preferences or needs. That is, Web Personalization becomes essential. Web Personalization [15] can be defined as the set of actions that are useful to adapt dynamically the presentation, the navigation schema and the contents of the Web, based on the preferences, abilities or requirements of the

user. Nowadays, as Brusilovsky et al. describe in [1], many research projects focus on this area, mostly in the context of e-Commerce [1] and e-learning [5].

We can distinguish three approaches for Web Personalization: the *manual decision rule approach*, the *content-based filtering approach* and the *collaborative filtering approach*. The first one requires the preconceived ideas of the experts about different kinds of users. Whereas the last two use information from previous navigations of the users and data mining techniques. Although the three approaches mentioned can be complementary, the first two can be too biased to the concrete user, too subjective and very often static. The third approach does not have the mentioned drawbacks but, however, it has the problem of scalability. One of the ways to make scalable a system based on collaborative filtering approach is to use data mining techniques to generate profiles in a batch process, or off-line, and then to use the outcome to accelerate the on-line personalization process [1].

The aim of the adaptation of Web sites according to the needs or preferences of the users will be to reduce at maximum the effort of the user. This paper presents a step in that direction that focuses on the design of a complete and generic system that can adapt the web pages to new users' navigation preferences proposing or emphasizing links that they will probably be using in a short future. We built a system and performed experiments based on a web-accessible database composed by server log information captured in the NASA, with 20K examples for training and 10K for testing but the procedure could be applied to any web environment. The system is based just on the information of server log files, the combination of a hierarchical clustering algorithm (SAHN) followed by an adaptation of a novel methodology to obtain the best partition from a cluster hierarchy (SEP/COP [6]) and a generalization step to detect user groups, and, SPADE [17] to obtain profiles for these groups.

We developed the described system and performed experiments to try to answer the following research question: is it possible to propose web adaptations that are useful for the users using just server log information? Is it always interesting to work with generalized URLs or is it worth maintaining the specificity of the URLs in some stages? Is the adaptation we propose for SEP/COP, so that it can be applied to sequences, useful for this application? Is SPADE useful to obtain the profiles of users with similar navigation behaviour? And finally, is the designed system able to solve the cold start problem?

The paper proceeds to contextualize the whole Web Mining process in Section 2. In Section 3 we describe the database we used in the process and Section 4 is devoted to describing the system we designed. The paper continues in Section 5 where we describe some results of the performed experiments. Finally, we summarize in Section 6 the conclusions and further work.

2 Web Mining

Web mining relates to the use of data mining in the web. This can be done with many different objectives: information retrieval and web search, link analysis, web crawling, structured data extraction, information integration, opinion

mining, web usage mining, etc. The nature of the analyzed information divides all those applications in three categories [10]. When the analyzed information is related to the content of the web pages, the process is called Web Content Mining. The process is called Web Structure Mining [3], when the used information is related to the web structure (pages and hyper-links). Finally, when the aim is to find use or interaction patterns in the Web that allow to model the users' behavior so that adequate recommendations or adaptations can be done, the process is called Web Usage Mining [12]. The work presented in this paper is a Web Usage Mining [16] application and as every web usage mining process it can be divided in three main steps: data acquisition and preprocessing [2], pattern discovery and analysis, and, exploitation.

The data acquisition and preprocessing phase is not straightforward, it requires different steps such as fusion of data from multiple log files, cleaning, user identification, session identification, path completion processes, etc. Machine learning techniques are mainly applied in the pattern discovery and analysis phase to find sets of web users with common web related characteristics and the corresponding patterns. And finally, the patterns discovered during the previous steps are used in the exploitation phase to adapt the system and make the navigation more comfortable for new users.

3 Database

In this work we have used a database from *The Internet Traffic Archive* [8] concretely NASA-HTTP (National Aeronautics and Space Administration) database [13,14]. The data contained in this database belongs to web server logs of user requests. The server was located at NASA Kennedy Space Center in Florida and logs were collected during two periods of time. The first set of logs was collected from 00:00:00 July 1, 1995 until 23:59:59 July 31, 1995, a total of 31 days. The second log was collected from 00:00:00 August 1, 1995 until 23:59:59 August 31, 1995, a total of other 31 days. The complete database contains 3,461,612 requests. The contained information is similar to the standardized text file format, i.e. Common Log Format which is the minimum information saved on a web server. Therefore, the system proposed in this work will use the minimal possible amount of information and, as a consequence, it will be applicable to the information collected in any web server.

4 Proposed System

Since we used NASA-HTTP database [13,14] the data acquisition phase has not been part of our work. We have designed the system starting from the data preprocessing step up to the exploitation phase.

4.1 Data Preprocessing

We preprocessed the log files to obtain information from different users and sessions. Before identifying user sessions, we filtered erroneous requests, image

requests, etc. that have not relationship with HTML pages since they could have been automatically generated by the server. As a consequence, the only requests we took into account for experimentation are the ones related to user clicks. In order to make this information more homogeneous we represented accesses to the same page with the same character sequence even if they were accessing to different zones of the page.

We performed the user identification based on IP addresses and we used an heuristic to identify sessions within a users' activity: we fixed the expire time of each session to 30 minutes [11]. Once the users and their activity layer were identified, we selected the most relevant sessions; the ones with higher level of activity. We selected the ones that had 6 or more clicks. After applying the whole data pre-processing process to NASA-HTTP database, the size of the database was reduced to 347,731 HTML requests and 31,853 sessions composed of at least 6 clicks.

This information could be processed and behavioral patterns extracted to obtain a vector representation be used in a machine learning algorithm. However, we propose to represent the information corresponding to each of the sessions as a sequence of clicks preformed in different URLs. Note that this representation focuses on the visited URLs and the order of these visits.

We further added a generalization step to this representation because our aim is to identify general navigation patterns, and having too specific paths in the used data, will make complicated to draw conclusions from the output of machine learning algorithms. The aim of the generalization step and to represent the URLs with a higher level of abstraction. Experimentation in a previous work with a smaller database showed this generalization to be efficient.

The generalization step consists on erasing a fraction of the segments from the right side of the path to diminish their specificity. For each one of the visited URLs, we obtained the length of the generalized URL based on next expression:

$$\max \{MinNSegment, \alpha * NSegments\} \quad (1)$$

Where $NSegments$ represents the number of segments separated by '/' appearing in the URL and α and $MinNSegment$ are parameters that can be varied depending on the structure of the site. $MinNSegment$ represents the minimum number of segments starting from the root an URL can have after the generalization step whereas α represents the fraction of the URL that will be kept in the generalized version. This generalization process will allow to work with the general structure of the site avoiding the confusion that too specific zones could generate. For the NASA database we instantiated $MinNSegment = 3$ and $\alpha = 0.5$.

4.2 Pattern Discovery and Analysis

Most commercial tools perform statistical analysis over the collected data but machine learning techniques are in general able to extract more knowledge from data. In this context clustering techniques have shown to be adequate to discover user profiles [15].

We used clustering to group into the same segment users that show similar navigation patterns and Edit Distance [7] as a metric to compare sequences. In order to avoid the need for parameter setting some clustering algorithms have, we used a well-known agglomerative hierarchical algorithm known as the Sequential Agglomerative Hierarchical Non-overlapping algorithm (SAHN) [9] combined with SEP/COP [6], a methodology we proposed in a previous work to automatically obtain the best partition from a cluster hierarchy. Hierarchical clustering algorithms provide a set of nested partitions called a cluster hierarchy. Since the hierarchy is usually too complex it is reduced to a single partition by using cluster validity indexes. We showed that the classical method is often not useful and we proposed SEP (Search in the Extended Partition Set), a new method that efficiently searches in an extended partition set. Furthermore, we proposed a new cluster validity index, COP (index that satisfies Context-independent Optimality and Partiality properties), since many of the commonly used indexes cannot be used with SEP. Experiments performed with 30 synthetic and 3 real datasets confirmed that SEP/COP is superior to the method currently used and furthermore, it is less sensitive to noise. We could say that this method is a self-regulated method, since it does not need any external parameter to obtain the best partition. For this work, we implemented and adaptation of COP index. COP index uses the average distance to the centroid to calculate the inter-cluster distance. In this application, since we are working with URL sequences, it is impossible to obtain the centroid of a cluster. As a consequence, we modified COP so that the intra-cluster distance is calculated as the average distance to the medoid. That is, the example with minimum average distance to the rest of the examples in the cluster.

The outcome of the clustering process will be a set of groups of user sessions that show similar behavior. But we intend to discover the associated navigation patterns for each one of the discovered groups. That is, common click sequences appearing among the sessions in a cluster. In this step we returned to the complete URLs and used SPADE (Sequential PAttern Discovery using Equivalence classes) [17], an efficient algorithm for mining frequent sequences, to extract the most common click sequences of the cluster. The application of SPADE provides for each cluster, a set of URLs that are likely to be visited for the sessions belonging to it. The number of the proposed URLs depends on parameters of the SPADE algorithm such as minimum support or maximum allowed number of sequences per cluster. The adequate value for the parameters depends on characteristics of the clusters such as size, structure or compactness, etc. that will vary from one to the other. There are different options to select the set of proposed URLs for each of the groups, but the analysis of the best option is out of the scope of this paper. As a consequence, we selected a fixed value for the minimum support and used it for all the clusters. After several experiments we decided 0.5 to be a good value.

The debate about the length of the proposed patterns stays open but in order to make the evaluation easier, in this approach we have only worked with rules of length one.

4.3 Exploitation

This is the part that needs to be done in real time. Up to now, we have identified groups of users navigating in similar areas and the URLs that are most likely to be visited, or most common paths, for each of the groups. When new users are navigating in the web site, the distance of their click sequence (single linkage, average linkage, distance to the medoid ... based on Edit distance [7]) to the clusters generated in the previous phase can be calculated. This can be done taking into account generalized URLs in the clusters and test examples, or going back to the original URLs. Our hypothesis is that the navigation pattern of that user will be similar to the navigation patterns grouped in its nearest cluster. Based on the most representative patterns obtained in that concrete cluster, the most interesting links can be proposed or outlined to the new user.

5 Experimental Results and Analysis

In order to evaluate the performance of the whole process, we divided the NASA database in two parts. One for training or generating the model, that is, for generating the clusters and extracting rules and another one for testing, that is, for evaluating to what extent the predictions made by the system would come along with the navigation performed in those sessions. Simulating a real situation we based the division of the database in temporal criteria: we used the oldest examples (66% of the database, 21,235 user sessions) for training and the latest ones (33%, 10,618 user sessions), for testing. In the training database, the total number of requests is 235,771 and the average number of clicks per session 11,1. The test database seems to have similar characteristics. Although it is smaller, the total number of requests is 111,960, the average number of clicks per session is similar to the one in the training sample: 10,5.

We applied the combination of SHAN and the modified SEP/COP to the training data so that the sessions with similar navigation characteristics are clustered into the same group. The outcome of the process contains 2,818 clusters where 726 clusters contain just a single session and 638 clusters have just two sessions. From this outcome we can conclude that some of the sessions belong to users with odd behavior and they have not been grouped with any other session. As the navigation patterns of these users do not seem to be very common, in order to build a system with smaller computational cost, we have desistimated them. We explored the behavior of the system using different values as minimum number of examples per cluster. Concretely, we experimented with 3, 5, 10, 20, 30, 40 and 50.

After the clustering process, we applied SPADE to generate the profile of each group of users. These profiles will be compared to the URLs in the test examples to evaluate the performance of the system.

To validate the system, we computed distances to find the most similar group for each of the test examples. We computed the distances for generalized URLs and for not generalized URLs and compared the results achieved with both versions. We performed this comparison taking into account the 0-day problem.

That is, although in the used database we have the complete navigation sequence of the test sessions, in real executions, when a user starts navigating, only its first few clicks will be available to be used for deciding the corresponding profile, and proposing new links according to it. We have simulated the real situation using 10%, 25% and 50% of the user navigation sequence, in order to select the nearest cluster or profile.

As a first approach, we obtained an upper bound performing the experiments for the complete test sequences (100 %) and a more realistic approach, performing the experiments with 50% of the URLs appearing in each test sequence (50 %) (see Table II). For each one of the options proposed in previous paragraphs, each test example and its nearest group of sessions, we used two points of view to evaluate the performance (see Table II):

Table 1. Average performance of the prediction systems using the whole test sequences (100%) and 50% of each test sequence (50%) with and without generalization

<i>MinClusterSize</i>	3	5	10	20	30	40	50	
<i>#clusters</i>	1454	847	359	156	89	57	45	
100.00%	Generalization							
	<i>precision</i>	66.5	69.8	73.8	71.5	70.3	69.8	69.2
	<i>recall</i>	29.7	28.5	27.4	25.2	23.2	22.4	22.2
	<i>%users touched</i>	92.2	92.7	92.7	90.5	88.2	87.5	86.5
	<i>F-measure</i>	53.29	54.12	55.13	52.29	50.0	49.04	48.62
	No Generalization							
	<i>precision</i>	71.1	74.1	75.1	72.4	71.1	69.2	68.1
	<i>recall</i>	35.9	34.8	32.2	29.1	26.4	25.0	24.8
	<i>%users touched</i>	93.4	97.5	98.9	97.4	96.2	95.2	94.0
	<i>F-measure</i>	59.44	60.45	59.30	55.80	53.11	51.12	50.47
50.00%	Generalization							
	<i>precision</i>	63.3	69.3	71.3	68.7	65.3	64.1	64.1
	<i>recall</i>	26.7	25.9	24.5	22.7	20.9	20.0	20.1
	<i>%users touched</i>	96.5	90.1	89.4	90.0	87.4	86.2	84.8
	<i>F-measure</i>	49.68	51.90	51.59	48.89	45.83	44.48	44.58
	No Generalization							
	<i>precision</i>	66.5	69.8	70.5	68.8	67.5	65.8	64.6
	<i>recall</i>	33.3	31.4	28.6	25.6	23.5	22.4	22.2
	<i>%users touched</i>	96.4	98.3	98.0	95.5	93.9	92.6	91.2
	<i>F-measure</i>	55.44	56.08	54.52	51.44	49.11	47.42	46.74

- We computed statistics based on results for each one of the new users. We compared the number of proposed links that are really used in the test examples (hits) and the number of proposals that are not used (misses) and calculated precision (*precision*), recall (*recall*) and $F_{0.5}$ -measure (*F-measure*). An ideal system would maintain precision and recall as high as possible. But in this kind of systems, since we can not expect to guess the whole navigation path of new users, it will be more important a high precision. This is why we used $F_{0.5}$ -measure to evaluate results.

- We counted the percentage of test examples that used at least one of the URLs proposed by the system (*% users touched*). That is, the percentage of users that would get some kind of satisfaction.

The first two rows in Table 1 summarize the general characteristics of the generated clusters: minimum number of examples per cluster (*MinClusterSize*) and the amount of generated clusters (*#clusters*). The rest of the rows summarize the results for the different options mentioned in previous paragraphs.

The first conclusion we can draw from this table is that even if the values of the measured parameters vary depending on the selected option, all of them are able to predict a certain percentage the links a new user will be visiting. As a consequence, we could claim that we designed a general system able to predict some of the links that a new user in the web will probably be using.

Analyzing all the results it seems that the option where the most similar group to the test example is found without generalization works better in every case and from the two points of view: $F_{0.5}$ -measure and number of users touched. This makes us think that for the prediction phase, the specificity of the accessed links is an important aspect.

Moreover, it seems that in the structure captured by the combination of SAHN and the modified SEP/COP, up to a point, not only large clusters are significant. Because, if we ignore them, for example, if we ignore the ones with less than 30 examples, the system performance worsens. Nevertheless, it is not clear which is the best minimum size for the generated clusters. Independently of the used criteria, $F_{0.5}$ -measure or number of users touched, it could be either 5 or 10 depending on the rest of parameters because the difference between the two options is small.

Table 2. Average performance of the prediction systems with minimum cluster size of 5 and 10 and without generalization when tackling the 0-day problem

		<i>UsedPercentage</i>	10%	25%	50%	100%
MinClusterSize10	<i>precision</i>		59.2	65.6	70.5	75.1
	<i>recall</i>		22.0	25.7	28.6	32.2
	<i>%users touched</i>		94.9	97.4	98.0	98.9
	<i>F-measure</i>		44.24	50.06	54.55	59.30
MinClusterSize5	<i>precision</i>		49.0	63.7	69.8	74.1
	<i>recall</i>		23.4	27.8	31.4	34.8
	<i>%users touched</i>		96.3	98.2	98.3	97.5
	<i>F-measure</i>		40.20	50.63	56.08	60.45

Finally, if we center the analysis in the 0-day problem, we realize that although the quality of results decreases when we use 50% of the URLs in each test sequence, the results are still good in this last case, obtaining precision values up to 71.3% and being at least one of the proposed URLs useful for up to 98.3% of the test examples. As we mentioned before, we went further in this analysis and evaluated how the system works when just 10% or 25% of the URLs in each test sequence is used to select the corresponding group. We show results of the

complete evolution fixing the rest of the parameters to: without generalization and minimum cluster sizes of 10 and 5 in Table 2.

If we analyze Table 2 we can observe that, although as it could logically be expected, the quality of the results decreases as we make the prediction at an earlier stage, results achieved with predictions made at very early stages (10%) are still good obtaining precision values up to 59.2% and being at least one of the proposed URLs useful for up to 96.3% of the test examples. So we could say that the proposed system is definitely able to tackle the 0-day problem.

6 Conclusions and Further Work

We designed a system that identifies different user profiles, and makes navigation proposals to new users using just server log information and machine learning techniques. This work has been done for, NASA-HTTP database [13, 14], but could be extended to any other environment. We preprocessed the data to identify users and sessions on the one hand, and prepared it so that it could be used with machine learning algorithms. We divided the database in two parts one for training and the other one for testing. We then proposed a generalization step with and applied clustering to the training data to discover groups of users with similar interests or navigation patterns. We used the self-regulated system composed by SAHN and SEP/COP, and proposed an adaptation to COP so that it can be used with sequences. We then used SPADE algorithm to discover the patterns associated to each of the clusters so the links that will be proposed to new users. We evaluated different configurations of the system and we also took into account the 0-day problem.

The validation results showed that the discovered patterns made sense and that they could be used to ease the navigation of future users by proposing or underlying links that they will probably use even if the prediction is made at very early stages of their navigation (10%). The best results were achieved when the nearest group is selected without generalization and limiting the minimum amount of examples of each cluster to 10 or 5. So, we could conclude that we have been able to design a generic system that, based only in web server log information, is able to propose adaptations to make easier and more efficient the navigation of new users. Since at this point we haven't used any domain specific information, this system would be useful for any web site collecting server log information.

In the future, this work could be improved in many senses. On the one hand it would be possible to use web structure information and content information of the selected web page for improving the results of the system. The outcome of this work could be applied to newer databases and it will be of great help when trying to adapt the Web to users with special needs. In that case, we will need to extract some more features from the database that will give us information about physical or cognitive characteristics of the users.

Acknowledgements. This work was funded by the University of the Basque Country, general funding for research groups (GIU10/02), by the Science and

Education Department of the Spanish Government (TIN2010-15549 project), by the Diputación Foral de Gipuzkoa and the FPI program of the Basque Government. We would also like to thank the people that did the job of collecting the logs: Jim Dumoulin of the Kennedy Space Center with contribution of Martin Arlitt (mfa126@cs.usask.ca) and Carey Williamson of the University of Saskatchewan.

References

1. Brusilovsky, P., Kobsa, A., Nejd, W.: The Adaptive Web: Methods and Strategies of Web Personalization. LNCS, vol. 4321. Springer, Heidelberg (2007)
2. Cooley, R., Mobasher, B., Srivastava, J.: Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems* 1(1) (1999)
3. Desikan, P., Srivastava, J., Kumar, V., Tan, P.N.: Hyperlink Analysis - Techniques and Applications. Army High Performance Computing Center Technical Report (2002)
4. EPA-HTTP logs. HTTP requests to the EPA WWW server located at Research Triangle Park, NC (1995), <http://ita.ee.lbl.gov/html/contrib/EPA-HTTP.html>
5. García, E., Romero, C., Ventura, S., De Castro, C.: An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. *User Modeling User and Adapted Interaction* 19(1-2), 99–132 (2009)
6. Gurrutxaga, I., Albisua, I., Arbelaitz, O., Martín, J.I., Muguerza, J., Pérez, J.M., Perona, I.: SEP/COP: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index. *Pattern Recognition* 43(10), 3364–3373 (2010)
7. Gusfield, D.: Algorithms on strings, trees, and sequences. Cambridge University Press (1997)
8. The Internet Traffic Archive, ACM SIGCOMM, <http://ita.ee.lbl.gov/>
9. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, Upper Saddle River (1988)
10. Kosala, R., Blockeel, H.: Web Mining Research: A Survey. *ACM SIGKDD Explorations Newsletter* 2(1), 1–15 (2000)
11. Liu, B.: Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data. Springer, Heidelberg (2007)
12. Mobasher, B.: Web Usage Mining. In: Web Data Mining: Exploring Hyperlinks, Contents and Usage Data. Springer, Berlin (2006)
13. NASA-HTTP logs. HTTP requests to the NASA Kennedy Space Center WWW server in Florida (1995), <http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html>
14. National Aeronautics and Space Administration (2010), <http://www.nasa.gov/>
15. Pierrakos, D., Paliouras, G., Papatheodorou, C., Spyropoulos, C.D.: Web Usage Mining as a Tool for Personalization: A Survey. *User Modeling and User Adapted Interaction* 13, 311–372 (2003)
16. Srivastava, J., Desikan, P., Kumar, V.: Web Mining - Concepts, Applications & Research Directions. In: Foundations and Advances in Data Mining. Springer, Heidelberg (2005)
17. Zaki, M.J.: SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning* 42, 31–60 (2001)

A New Criteria for Selecting Neighborhood in Memory-Based Recommender Systems*

Sergio Cleger-Tamayo¹, Juan M. Fernández-Luna², and Juan F. Huete²

¹ Departamento de Informática, Facultad de Informática y Matemática
Universidad de Holguín, 80100, Holguín, Cuba
`sergio@facinf.uho.edu.cu`

² Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática y de Telecomunicación, CITIC – UGR,
Universidad de Granada, 18071, Granada, Spain
`{jmfluna@decsai, jhg}@decsai.ugr.es`

Abstract. In this paper a new proposal for memory-based Collaborative Filtering algorithms is presented. In order to compute its recommendations, a first step in memory-based methods is to find the neighborhood for the active user. Typically, this process is carried out by considering a vector-based similarity measure over the users' ratings. This paper presents a new similarity criteria between users that could be used to both neighborhood selection and prediction processes. This criteria is based on the idea that if a user was good predicting the past ratings for the active user, then his/her predictions will be also valid in the future. Thus, instead of considering a vector-based measure between given ratings, this paper shows that it is possible to consider a distance between the real ratings (given by the active user in the past) and the ones predicted by a candidate neighbor. This distance measures the quality of each candidate neighbor at predicting the past ratings. The best-N predictors will be selected as the neighborhood.

Keywords: Filtering and Recommending, Collaborative Filtering.

1 Introduction

Recommender Systems (RSs) tries to help users either to find what they are explicitly looking for or what they would find useful from a vast amounts of information. There are a large number of RSs currently in Internet sites as, for example, Amazon, Netflix or Jester, that demonstrates the increasing interest in this technology for reducing information overload. In few words, these systems suggest items to the user (according to his/her context) [6, 13], representing a step ahead in the context of traditional Information Retrieval.

The main task normally associated to RSs is *rating prediction*, i.e. the RS estimates how the user would rate a given item. Traditionally, in RSs there

* This work has been jointly supported by the Spanish Ministerio de Ciencia e Innovación, under project TIN2008-06566-C04-01, and the Consejería de Innovacion, Ciencia y Empresa de la Junta de Andalucía under project P09-TIC-4526.

exists a number m of items or products $I = \{I_1, I_2, \dots, I_m\}$, a group of n users, $U = \{U_1, U_2, \dots, U_n\}$ and for each user, that can be denoted by U or V , a set of ratings for those observed items in I , being $r_{u,i}$ the rating given by user U to the item I . Usually, these ratings are given using a value from a set S of possible ratings (for instance, $S = \{1, \dots, 5\}$ or $S = \{\text{like}, \text{dislike}\}$).

On the one hand, we want to highlight the active user, U_a or A , which is the one that is interacting with the system, and on the other hand, we also highlight an item I_t , the target item, that will be the one for which we are making predictions, denoted as $\hat{r}_{a,t}$. In order to perform the predictions, a RS needs a model of the users' preferences or tastes (user profile) which is usually learned from how the user rates those observed items in the past. RSs are usually grouped into two categories [1]: *Content-based Recommenders* make recommendations based on the user preference model that combines the user's ratings with content description of the items. *Collaborative filtering systems* [13] use the ratings of like-minded users to make recommendations for a given user.

Due to its good performance, there exists a great interest in collaborative filtering (CF). According to [3], collaborative RSs can be grouped into memory-based and model-based approaches. The first, also known as neighborhood-based approaches, use the entire rating matrix to make recommendations, while model-based algorithms predictions are made by building an explicit model of the user preferences. Then, this model is used to predict the target ratings.

The idea behind neighborhood-based approaches is that if two users rated similarly, then they can be used for prediction purposes. We want to note that selecting the right users is important because they are the ones used to compute the predictions. Many empiric studies and real-world systems in the literature have employed traditional vector similarity measures (say Pearson's correlation in different formulations, cosine, Spearman's Rank, etc. [8, 7, 14, 11, 2, 5, 16]). These similarities measure the closeness between users' ratings but, as it is well known, they are not apt to completely characterize their relationship.

In this paper we shall explore a new neighborhood-based approach to perform the recommendation process which is based on predictive criteria. Particularly, and in order to find the neighborhood for the active user, we shall consider how good is a candidate neighbor if he/she were used to predict the rating given by the active user to his/her observed items, i.e. his/her past rating. So, in few words, we shall consider predictive capability instead of rating similarities.

This paper is organized in the following way: the next section presents related work on memory-based recommender systems. Then Section 3 presents the motivation of our approach. An empirical experimentation with MovieLens datasets is presented in Section 4. Finally, Section 5 presents the concluding remarks.

2 Related Work: Neighborhood-Based Recommender Systems

In order to compute the predictions, memory-based RS can be classified into: user-based methods [8, 15] that use an aggregation measure which considers the

ratings given by similar users for the same item and item-based approaches [14,4], taking into account the similarity between items (two items are similar if they have been rated similarly) and the predictions are generated considering those ratings given by the active user to similar items.

In this section we shall outline the works carried out by other authors when considering an user-based approach. In this sense, three main tasks will be considered: firstly, how to determine the neighborhood and secondly how to aggregate the ratings given by similar users to the target item.

2.1 Neighborhood Selection

In order to compute recommendation for the target item I_t , only the users that have rated this item will take part in the neighborhood, the idea is to select the best among them, denoted as $N_t(a)$. Herlocker et. al. in [7], demonstrated that it is better to set a fixed number of neighbors (in a range from 20 to 60) than using a threshold on the similarity weights. The most common similarity metrics in user-based RS [5] are:

- Pearson Correlation Coefficient (PC): This metric measures the degree of association between the ratings patterns using a value between -1 and +1.
- Cosine Measure: This metric defines the similarity between two users as the cosine angle between the rating vectors, with values between 0 and 1. A larger value means that the similarity of the ratings increases.
- Mean Square Difference: This is a distance measure (but the inverse can be considered as a similarity metric) which evaluates the distance between two users as the average squared difference between the rating given by the user to the same items.

Different empirical analysis have demonstrate that Pearson’s correlation obtains better neighborhood [3,8,7,10]. Moreover, better performance is obtained when devaluing the correlations that are based on small numbers of co-rated items, k :

$$sim(U, V) = PC(U, V) \cdot CF, \tag{1}$$

with $CF = 1$ if $k > 50$ and $CF = k/50$ otherwise.

Finally, the best- n neighbors having larger sim values are used.

2.2 Computing the Predictions

As we say in Section 1 and 3, the methods mentioned in the literature to compute the predicted rating, $\hat{r}_{a,t}$, use the ratings given to I_t by the best- n neighbors.

A user-based rating prediction can be formalized as an aggregation of the ratings that the different neighbors suggest to the target item, denoted by $f_A(V, t)$. These suggestions are combined by weighting the contribution of each neighbor by its similarity with respect to the active user (check [5] for a good review):

$$\hat{r}_{a,t} = \frac{\sum_{V \in N_t(a)} sim(A, V) f_A(V, t)}{\sum_{V \in N_t(a)} sim(A, V)}. \tag{2}$$

Different approaches can be considered in the literature which mainly differ on how the neighbors' suggestions are computed [7,10]. The selection of one of them highly depends on the features of the rating matrix, R . We will present the three main approaches:

1. Raw ratings. It is considered that users suggest his/her own rating given to the target item, i.e. $f_A(V, t) = r_{v,t}$. Then,

$$\hat{r}_{a,t} = \frac{\sum_{V \in N_t(a)} \text{sim}(A, V) r_{v,t}}{\sum_{V \in N_t(a)} \text{sim}(A, V)}. \quad (3)$$

2. Mean-centering suggestion. We are considering that users may use a different rating scale to quantify the same level of preferences for an item. Therefore, the suggestion is obtained by considering the difference between the rating and the mean rating, i.e. $f_A(V, t) = \bar{r}_a + (r_{v,t} - \bar{r}_v)$. So, we can obtain the popular prediction approach in [7], i.e.

$$\hat{r}_{a,t} = \bar{r}_a + \frac{\sum_{V \in N_t(a)} (r_{v,t} - \bar{r}_v) \text{sim}(A, V)}{\sum_{V \in N_t(a)} \text{sim}(A, V)}. \quad (4)$$

3. Z-score-based suggestion. Additionally to the difference in the rating scale, this criterion also considers the variance in the individual rating scales. Thus, the neighbors' suggestions are computed by dividing the mean-centered rating by the standard deviation σ , i.e. $f_A(V, t) = \bar{r}_a + \sigma_a (r_{v,t} - \bar{r}_v) / \sigma_v$. So, the predicted rating is computed using the following equation:

$$\hat{r}_{a,t} = \bar{r}_a + \sigma_a \frac{\sum_{V \in N_t(a)} (r_{v,t} - \bar{r}_v) / \sigma_v \cdot \text{sim}(A, V)}{\sum_{V \in N_t(a)} \text{sim}(A, V)}. \quad (5)$$

3 A New Metric for Selecting the Neighborhood

In order to illustrate our approach we shall consider the set of ratings in Table 1 with the objective of computing Ann's prediction for the target item I_t , for instance assume that $I_t = I_7$.

Following [8,7], neighborhood-based methods, the first step to find the neighborhood is to weight all users/items with respect to the selected similarity criterion. Thus, considering the Pearson's correlation (PC) coefficient, it can be found that the three candidates have the same value of PC, i.e. $PC(Ann, Sara) = PC(Ann, John) = PC(Ann, Bill) = 1$. So, we can not distinguish between these candidate users.

As we have seen, in the prediction processes each candidate user suggests a rating $f_A(V, t)$, which is combined to obtain the final prediction, see eq. 2. Thus, independently of the criterion used to compute $f_A(U, t)$, each user (Sara, John or Bill) will suggest a different rating. For instance, if $f_A(V, t) = \bar{r}_a + (r_{v,t} - \bar{r}_v)$ we have that $f(Sara, I_t) = 2.42$, $f(John, I_t) = 3.21$ and $f(Bill, I_t) = 5.47$.

The problem that arises is: Which is the best prediction?, or in other words, Which user may help us to make better predictions?. In this paper we are going

Table 1. Measuring user similarities. The objective is to provide Ann’s prediction for a target item I_t .

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	\dots	I_{18}
Sara	5	4	5	3	3	3	2	2	2	2	2	\dots		2
John	4	3	4	2	2	1	2	1	1	1	1	1	\dots	1
Bill	3	2	3	1	1	1	5	1	3	3	3	3	\dots	3
Ann	4	3	4	2	2		?							

to explore this situation from a new perspective: as an alternative to measure the similarities between users.

Focusing on this objective, the idea will be to measure the capability of each user to predict Ann’s past ratings. In this case, using the data in Table 1, we can focus on item I_1 which was rated with 4 by Ann. We might wonder how it might rate Sara’s suggestion for this particular item. Thus, using $f_{Ann}(Sara, I_1) = \bar{r}_{Ann} + (r_{Sara, I_1} - \bar{r}_{Sara})$ we have that the expected prediction is $f(Sara, I_1) = 3 + (5 - 2.58) = 5.42$. Similarly, using John’s suggestions, $f_{Ann}(John, I_1) = 5.21$ and using Bill’s suggestion we might expect the rating $f_{Ann}(Bill, I_1) = 3.47$. So, the closer suggestion with respect to the real rating is the one given by Bill. This idea will be the basis of a new metric to measure the quality of a neighbor.

Our hypothesis is that if a user U was good at predicting the active user’s past ratings, then his/her prediction for an unobserved item will also be good. Note that the key aspect in this approach is that we shall consider the rating’s predictions instead of the raw ratings. Therefore, we shall have, on the one hand, the ratings given by the active user, $R_A = \{r_{a,1}, \dots, r_{a,m}\}$, to his/her m rated items and, on the other hand, the ratings that will be suggested (predicted) for each item by the user U , denoted as $\hat{F}_A(U) = \{\hat{r}_{a,1}, \dots, \hat{r}_{a,m}\}$.

In order to determine the quality of the predictions, which measures in some sense the predictive capability of a user, we have to consider if the predicted ratings, $\hat{F}_A(U)$, and the original ones, R_A , are similar in absolute terms. In this paper, we propose the use of a loss function to measure the magnitude of difference between the two ratings, $L(r_{a,i}, \hat{r}_{a,i})$, representing the cost incurred between the true rating $r_{a,i}$ and user U predicted rating $\hat{r}_{a,i}$. Particularly, we shall consider the average absolute deviation between a predicted rating, $\hat{r}_{a,i}$, and the user’s true rating, $r_{a,i}$:

$$L(r_{a,i}, \hat{r}_{a,i}) = abs(r_{a,i} - \hat{r}_{a,i}). \tag{6}$$

We would like to note that it is not possible to use Pearson Correlation or Cosine measures for this purpose, although they both measure the closeness between ratings but, as it is well known, high correlations with the predictions does not necessarily imply having good predictions. For instance, we can imagine the situation where the prediction is $r + k$ with correlation equals to one, but the performance worsened with k .

Therefore, the predictive capability of a user can be measured by considering the expected loss over the active user’s past ratings. Note that similarly to those vector based approaches, the expected loss will be computed taking into account

the common ratings, i.e. those ratings in $R_A \cap R_U$, being R_\bullet the subset of items rated by the user, i.e.

$$EL(A, U) = \frac{\sum_{i \in R_A \cap R_U} L(r_{a,i}, \hat{r}_{a,i}(u))}{|R_A \cap R_U|}. \tag{7}$$

Since we are considering a memory-based approach for recommending a rating for the target item, I_t , we have to compute the $EL(A,U)$ for those users, U , who rated the target item I_t , i.e. U such that $r_{u,t} \in R_U$. These users will be ranked in increasing order of their expected loss and the top N will be selected as the active user’s neighbors. Then, their suggested ratings will be combined in such a way that the best predictors will have a greater weight in the combination. To conclude, the algorithm in Table 2 summarizes our recommendation model.

Table 2. Based-Correlation Predictive Model

Inputs: A active user, I_k target item
Output: r_a predicted rating
1. Neighbors Selection 1.1 For each U who rated the target item I_t 1.1.1 Compute the $EL(A, U)$, $EL(A, U) = \frac{\sum_{i \in R_A \cap R_U} L(r_{a,i}, \hat{r}_{a,i}(u))}{ R_A \cap R_U }$ 1.2 Rank the users and select the best- N neighbors, i.e. those with lower E.L 2. Predictive Process 2.1 Compute a rating $r_{a,t}$ according to $\hat{r}_{a,t} = \frac{\sum_{V \in N_t(a)} EL(A,V) f_A(V,t)}{\sum_{V \in N_t(a)} EL(A,V)}.$

From the efficiency perspective we have to say that our approach is equivalent to vector-based measures since the predictions can be computed in constant time, assuming that some statistics as the user mean rating or the standard deviation have been previously computed.

4 Evaluation of the Recommender Model

This section establishes the evaluation settings and also presents the experimental results for the performance of the model.

4.1 Data Set and Experimental Methodology

In terms of the test data set, we have decided to use MovieLens [12]. It was collected by the GroupLens Research Project at the University of Minnesota and contains 100,000 anonymous ratings (on a scale of 1 to 5) of approximately 1,682 movies made by 943 MovieLens users who joined MovieLens during the seven-month period from September 19th, 1997 through April 22nd, 1998.

The objective of our experimentation is to measure the capability of the system at predicting the interest of the user for an unobserved item, i.e. we shall consider the task of rating prediction. In order to validate our model, and similarly to most machine learning evaluation methodologies, we shall randomly divide our data sets into training (containing 80% of the ratings) and test set (containing 20%). To reduce variability on the results, we run 5-fold cross-validation over different partitions. The results presented in this experimentation are the average values obtained over the five rounds.

In order to test the performance of our models, we shall measure how well the system predicts the user's true ratings or preferences, i.e. the system accuracy. Following [9], we propose to use three different evaluation metrics Hamming, MAE and RMSE which basically differs in the loss associated to each error. Thus, the first one measures the number of correct predictions obtained, the second one considers the Mean Absolute Error and the last one the Root Mean Squared Error, RMSE, which squares the errors before they are averaged, given a relatively high weight to large errors. This metric is most useful when large errors are particularly undesirable. In order to obtain the error in the same magnitude as the ratings, the root of the expected loss is computed.

Finally, we want to say that two different alternatives have been studied to select the neighborhood: the first one requires that in order to do the recommendations, the candidate users must have at least ten past ratings in common with the active user and, the second one, where this restriction is relaxed, requiring only one item in common.

4.2 Experimental Results

In order to illustrate the performance of our model we shall consider two different alternatives to learn the neighborhood, the one proposed in [7] as baseline (*BL*), where the best neighbors are obtained using Pearson Correlation and the one where the neighbors are selected using the EL over predicted ratings (eq. [7]). Also, to compute the rating prediction we shall consider the raw rating (eq. [3]) (*Raw*) and the normalized rating suggestions using the mean centering (eq. [4]) (*NormR*), the last has been proved to give good results. Note that the combination *BL-NormR* is the model proposed in [7] which remains as state-of-art model in memory-based collaborative filtering [10, 5].

Finally, and related to the size of the neighborhood, we have opted to consider a fixed number of neighbors. Particularly, we have used the best 5, 10, 20, 30 and 50 users.

Tables [3] and [4] show the performance of our metric when requiring 10 and 1 past ratings in common, respectively. In this sense, we have used a weight to devalue similarities that are based on a small numbers of co-rated items, as the CF presented in equation [1]. We highlight in boldface the best result for each experiment.

Several conclusions can be drawn from these results: Firstly, the best results have been obtained in all the experiment when considering the mean-centering based suggestions. Secondly, the size of the neighborhood has a significant impact

Table 3. Performance metrics with at least 10 common ratings

RMSE				
Neighbors	Raw	NormR	BL-Raw	BL-NormR
5	1,16749	1,10231	1,15479	1,09998
10	1,12701	1,06270	1,12668	1,07184
20	1,10176	1,04229	1,11382	1,05874
30	1,09458	1,03753	1,11124	1,05751
50	1,09143	1,03450	1,11319	1,05750
Hamming				
5	0,64772	0,61792	0,62930	0,61414
10	0,63413	0,6044	0,62051	0,60283
20	0,62888	0,59439	0,61876	0,59550
30	0,62726	0,59267	0,61938	0,59460
50	0,62711	0,59155	0,62199	0,59384
MAE				
5	0,85872	0,79544	0,82851	0,78397
10	0,82223	0,76066	0,8042	0,75786
20	0,80297	0,74062	0,79519	0,74457
30	0,79731	0,73661	0,79402	0,74301
50	0,79531	0,73391	0,79745	0,74218

Table 4. Performance metrics with at least 1 common rating

RMSE				
Neighbors	Raw	NormR	BL-Raw	BL-NormR
5	1,14236	1,08232	1,12510	1,07305
10	1,09832	1,03733	1,08319	1,03235
20	1,07205	1,01638	1,06364	1,01324
30	1,06560	1,01160	1,05865	1,00746
50	1,06347	1,00865	1,05650	1,00402
Hamming				
5	0,6464	0,61836	0,63809	0,6237
10	0,63113	0,60448	0,62742	0,60901
20	0,62683	0,59466	0,62173	0,60051
30	0,62666	0,59285	0,62097	0,59622
50	0,62654	0,5915	0,62164	0,59464
MAE				
5	0,84489	0,78695	0,82666	0,78524
10	0,80550	0,74912	0,79271	0,74995
20	0,78668	0,72933	0,77678	0,73208
30	0,78248	0,72501	0,77317	0,72543
50	0,7810	0,72204	0,77277	0,72203

on the quality of the results. The accuracy measures improve as we increase the neighborhood size from 5 to 50. These differences are statistically significant using the paired Wilcoxon test. Finally, it seems that requiring only one past

item in common allows us to obtain better neighborhood with both approaches. This demonstrates that in order to be a good neighbor it is not necessary to share many common movies with the active user. Although this is true, in the case of Hamming distance the differences are not significant.

As a global conclusion we can say that our criteria to select the neighborhood is competitive with the standard Pearson Correlation criteria in terms of both efficiency and efficacy (there is no statistical significance between the best results in Table 4, using the paired Wilcoxon test). We obtain similar results, but with a different neighborhood. This fact confirms our hypotheses that measuring the predictive capability might be helpful for predictive purposes. We want to say that this is a preliminary experimentation, so there exists room for further improvements.

5 Conclusions and Future Work

In this paper, we show the feasibility of building a memory based CF system that considers those users having greater impact in the (past) ratings of the active user. The performance is improved by taking into consideration how these users influence in the ratings of items similar to the target one.

As future work we plan to study the use of trust measures in order to increment the performance. Also, more experiments will be conducted with other collections in order to test the variability of the results with datasets presenting different features.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17(6), 734–749 (2005)
2. Ahn, H.J.: A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Inf. Sci.* 178(1), 37–51 (2008)
3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering, pp. 43–52. Morgan Kaufmann (1998)
4. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.* 22(1), 143–177 (2004)
5. Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: *Recommender Systems Handbook*, pp. 107–144 (2011)
6. Goldberg, D., Nichols, D.A., Oki, B.M., Terry, D.B.: Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35(12), 61–70 (1992)
7. Herlocker, J., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.* 5, 287–310 (2002)
8. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: *SIGIR*, pp. 230–237 (1999)
9. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 5–53 (2004)

10. Howe, A.E., Forbes, R.D.: Re-considering neighborhood-based collaborative filtering parameters in the context of new data. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, pp. 1481–1482. ACM, New York (2008)
11. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing* 7(1), 76–80 (2003)
12. Movielens: Data sets, <http://www.grouplens.org/>
13. Resnick, P., Varian, H.R.: Recommender systems. *Commun. ACM* 40, 56–58 (1997)
14. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, WWW 2001, pp. 285–295. ACM, New York (2001)
15. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating "word of mouth", pp. 210–217. ACM Press (1995)
16. Wang, J., de Vries, A.P., Reinders, M.J.T.: Unified relevance models for rating prediction in collaborative filtering. *ACM Trans. Inf. Syst.* 26, 16:1–16:42 (2008)

Extended Precision Quality Measure for Recommender Systems

Fernando Ortega, Antonio Hernando, and Jesús Bobadilla

Universidad Politecnica de Madrid & FilmAffinity.com research team
fortegarequena@gmail.com, ahernando@eui.upm.es,
jesus.bobadilla@upm.es

Abstract. Recommender systems are highly sensitive to cases of false-positives, that is, recommendations made which have proved not to be relevant. These situations often lead to a loss of trust in the system by the users; therefore, every improvement in the recommendation quality measures is important. Recommender systems which admit an extensive set of values in the votes (usually those which admit more than 5 stars to rate an item) cannot be assessed adequately using precision as a recommendation quality measure; this is due to the fact that the division of the possible values of the votes into just two sets, relevant (true-positive) and not-relevant (false-positive), proves to be too poor and involves the accumulation of values in the not-relevant set. In order to establish a balanced quality measure it is necessary to have access to detailed information on how the cases of false-positives are distributed. This paper provides the mathematical formalism which defines the precision quality measure in recommender systems and its generalization to extended-precision.

Keywords: Precision, quality measure, recommender systems, collaborative filtering.

1 Introduction

In recent years, Recommender Systems (RS) have played an important role in reducing the negative impact of information overload on those websites where users have the possibility of voting for their preferences on a series of articles or services. Movie recommendation websites are probably the most well-known cases to users and are without a doubt the most well studied by researchers [1-3], although there are many other fields in which RS have great and increasing importance, such as e-commerce [4] and e-learning [5,6].

RS make it possible for each user who uses the system to obtain the most relevant information in a personalised way. Conceptually, the way they work is very simple; a filtration process is performed for items using one of the following models:

- Content-based filtering [7,2] base the recommendations made to a user on the choices this user has made in the past (e.g. in a web-based e-commerce RS, if the user purchased some fiction films in the past, the RS will probably recommend a recent fiction film that he has not yet purchased on this website).

- Demographic filtering [8] based RS are based on the principle that individuals with certain common personal attributes (sex, age, country, etc.) will also have common preferences.
- Collaborative Filtering (CF) based RS [9,10] allow users to give ratings about a set of elements (e.g. videos, songs, films, etc. in a CF based website), in such a way that when enough information is stored on the system we can make recommendations to each user based on information provided by those users we consider to have the most in common with them.
- The RS hybrid user models [11-13] commonly use a combination of CF with demographic filtering or CF with content based filtering, to exploit merits of each one of these techniques.

Currently, Collaborative Filtering (CF) is the most commonly used and studied technology [9,10]. CF RS are based on the way in which humans have made decisions throughout history: besides on our own experiences, we also base our decisions on the experiences and knowledge that reach each of us from a relatively large group of acquaintances. We take this set of knowledge and we assess it “in a critical way”, to obtain the decision that we consider most appropriate in order to achieve a purpose. A key factor in the quality of the recommendations obtained in a CF based RS lies in its capacity to determine which users have the most in common (are the most similar) to a given user. A series of algorithms [14] and metrics [9,15,16,17,18] of similarity between users are currently available which enable this important function to be performed in the CF core of this type of RS.

Up to now, several publications have been written dealing with the way in which the RS are evaluated. Among the most significant we consider [10], which reviews the key decisions in evaluating CF RS: the user tasks, the type of analysis and datasets being used, the ways in which prediction quality is measured and the user-based evaluation of the system as a whole. The paper [19] is a current study which proposes a recommendation filtering process based on the distinction between interactive and non-interactive subsystems. There are also general publications and reviews including the most commonly accepted metrics, aggregation approaches and evaluation measures: mean absolute error, coverage, precision, recall and derivatives of these: mean squared error, normalized mean absolute error, ROC and fallout; [20] focuses on the aspects not related to the evaluation, [21] compare the predictive accuracy of various methods in a set of representative problem domains. [22, 23] review the main CF methods proposed in the literature.

The rest of the paper is divided into sections ordered according to the level of abstraction of the concepts they cover. Each numerical value on the list indicates its corresponding section number.

2. Motivation: why it is important the proposed extended-precision?
3. Extended precision concept details.
4. Mathematical formalism defining the proposed extended precision and the necessary collaborative filtering equations on which this quality measure is applied.
5. Design of the experiments with which to test the extended precision. The database FilmAffinity is used.

6. Graphical results complemented with the explanations on the behavior of each experiment.
7. Most relevant conclusions obtained and related work.

2 Motivation

The precision measure is important in the field of information retrieval; precision is the fraction of the documents retrieved that are relevant to the user's information need. In the field of RS the precision means the proportion of the recommendations made proven to be relevant: that is, the number of true-positives as regards the total set of recommendations made.

In short, the purpose in the CF stage of a RS deals with trying to maximize the number of relevant recommendations (and therefore, the precision), whilst trying to minimize the number of non-relevant recommendations. Both objectives are important to ensure that the RS is both useful and, at the same time, its users trust its results (recommendations).

The quality measures with which the RS are put to the test can be divided into:

- Prediction quality measures: accuracy, percentage of perfect predictions and coverage.
- Recommendation quality measures: precision, recall, fall-out, specificity, etc.

Of these measures, accuracy is almost definitely the most widely studied, by calculating the Mean Absolute Error (MAE), and precision (usually combined with recall) is the one which best indicates the integrity of the recommendations.

Despite the importance of precision as a recommendation quality measure in RS, no specific variation of it is used that would enable us to highlight the proportion of non-relevant recommendations. This situation is most likely due to the fact that most of the RS in operation have a possible range of votes in the interval $\{1, \dots, 5\}$, and therefore, only the "relevant" and "not-relevant" categorizations are used; this way, the precision indicates the proportion of relevant recommendations and the value $(1 - \text{precision})$ indicates the proportion of non-relevant recommendations.

Not all RS have a possible range of 5 or less values in the votes, and therefore, many situations exist in which it would be very useful to have access to detailed information on how the cases of false-positives are distributed, which would be grouped in the imprecise value $(1 - \text{precision})$. By way of example, in an RS in which votes can be cast in the range $\{1, \dots, 10\}$, we can establish as relevant values $\{8, 9, 10\}$, and after processing the database we can determine that the precision is 0.6 for $N=20$ (Number of recommendations). At this point, the question arises as to how the remaining $(1 - 0.6)$ is distributed, as it would be very acceptable for the 8 cases of false-positives to have value 7 (7 stars) and very unacceptable for the 8 cases to have values 1 or 2.

3 Extended Precision

The precision in RS is calculated based on the confusion matrix specified in Table 1, obtaining the following proportion:

$$precision = \frac{\#TruePositive}{\#TruePositive + \#FalsePositive} = \frac{\#TruePositive}{\#Recommended} = \frac{\#TruePositive}{N}$$

Table 1. Confusion matrix

	Relevant	Not relevant
Recommended	True-positive	False-positive
Not recommended	False-negative	True-negative

The precision formulated in this way refers to the capacity to obtain relevant recommendations regarding the total number of recommendations made (N). This value could be formulated as follows:

$$precision_{relevant} = \frac{\#True\ positives}{N}$$

This way, we could determine the precision value of the false-positive cases as:

$$precision_{not\ relevant} = \frac{\#FalsePositive}{N} = 1 - precision_{relevant}$$

In these situations, there is no sense establishing as independent objectives an increase in the $precision_{relevant}$ and a reduction in the $precision_{not\ relevant}$, as an inverse linear relationship exists between the two, and therefore, only one of them is ever calculated ($precision_{relevant}$).

In RS in which there is a wide number of possible values of votes, Table 1 can be extended with new values of relevance for the votes, as shown, by way of example, in Table 2. In this case, we could have established the following sets:

$$very\ relevant = \{9,10\}, relevant = \{7,8\}, slightly\ relevant = \{5,6\}, not\ relevant = \{1,2,3,4\}$$

Table 2. Extended confusion matrix

	Very relevant	Relevant	Slightly relevant	Not relevant
Recommended	True-positive	False-positives		
Not recommended	False-negative	True-negatives		

As $precision_{not\ relevant} \neq 1 - precision_{relevant}$, it is appropriated to distinguish between each different case into which the false-positives and the true-positive case are grouped:

$$precision_{very\ relevant} = \frac{\#Very\ relevant}{N}, precision_{relevant} = \frac{\#Relevant}{N},$$

$$precision_{slightly\ relevant} = \frac{\#Slightly\ relevant}{N}, precision_{not\ relevant} = \frac{\#Not\ relevant}{N}$$

We will now be capable of observing the integrity of the recommendations made by analyzing the values of each of the groups established; this information can be combined with the limit requirements that we establish in our RS (e.g. minimum value accepted in *precision_{very relevant}* and maximum value accepted in *precision_{not relevant}*).

By using the traditional precision measure, the quality determination approach is restricted to: a) establishing a number of recommendations (N), b) calculating the precision value, and c) deciding whether that value exceeds the quality threshold set for the RS. Using extended precision, it is possible to establish varied quality conditions which adapt to the specific minimum quality requirements that we wish to set in the RS.

By way of example, we can establish the following quality conditions:

1. $Precision\ Quality = N > 20\ AND\ N < 40\ AND\ Precision_{very\ relevant} > 0.3\ AND\ Precision_{relevant} > 0.48\ AND\ Precision_{not\ relevant} < 0.2$
2. $N = Precision_{very\ relevant} > 0.3\ AND\ Precision_{relevant} > 0.48\ AND\ Precision_{not\ relevant} < 0.2$
3. $N = (Precision_{very\ relevant} > 0.3\ OR\ Precision_{relevant} > 0.42)\ AND\ Precision_{not\ relevant} < 0.22$

In the first case we can see whether the condition is met, that is, whether there are any values in the set interval ($N \in \{20, \dots, 40\}$) which meet the 3 quality requirements specified. In the second and third cases we obtain the ranges of values of N in which the condition is met.

Using the extended-precision approach proposed, the use of a query language enables the administrators of the RS to establish quality conditions for the recommendations and to obtain ranges of numbers of possible recommendations to users, maintaining the quality conditions established.

4 Formalization

Given an RS with a database of q users and m items rated in the range $[min..max]$, where the absence of ratings will be represented by the symbol \bullet .

Let U and I the sets of the RS users and items; $r_{u,i} = v$ the vote v of user u on item i . $p_{u,i} = p$ the value of the prediction p made to user u on item i . K_u the set of K users (neighbors) similar to active user u .

We define X_u as the set of recommendations to user u , and Z_u as the set of N recommendations to user u .

The following must be true:

$$X_u \subset I \wedge \forall i \in X_u, r_{u,i} = \bullet, p_{u,i} \neq \bullet, \quad (1)$$

$$Z_u \subseteq X_u, \#Z_u = N, \forall x \in Z_u, \forall y \in X_u : p_{u,x} \geq p_{u,y} \quad (2)$$

If we want to impose a minimum recommendation value: $\theta \in Real\ numbers$, we add $p_{u,i} \geq \theta$

We will divide the set of possible votes on one item, except empty ($V - \{\bullet\}$) into g subsets $L_j, \{L_j \mid j \in \{1, \dots, g\}\}$ (3)

Where j is the indicator of relevance of the values of each subset L_j , in such a way that L_1 will represent the votes with the least relevance and L_g will represent the votes with the most relevance. E.g.:

$$V = \{v \in \text{Natural numbers} \mid 1 \leq v \leq 10\}, g = 3, L_1 = \{1, 2, 3\}, L_2 = \{4, 5, 6, 7\}, L_3 = \{8, 9, 10\}$$

We will define the precision obtained for user u on the set of votes of relevance j as t_u^j , and the total precision obtained in the RS on the set of votes of relevance j as t^j .

$$t_u^j = \frac{\#\{i \in Z_u \mid r_{u,i} \in L_j\}}{N}, \text{ where: } \sum_{j=1}^g t_u^j = 1, t^j = \frac{1}{\#U} \sum_{u \in U} t_u^j \quad (4)$$

5 Design of Experiments

To carry out the experiments we used the *FilmAffinity.com* database, which, in the version used, consists of 26,447 users, 21,128 items, 19,126,278 ratings and a range of ratings from 1 to 10. We find the values of precision: very relevant, relevant, slightly relevant and not relevant, for a range of recommendations $N \in \{10, \dots, 100\}$, $K=180$, 20% of test users, 20% of test items.

The four experiments designed are composed of the sets shown in Table 3:

Table 3. Experiments designed

	Very relevant	Relevant	Slightly relevant	Not relevant
Experiment <i>a</i>	{9,10}	{7,8}	{5,6}	{1,2,3,4}
Experiment <i>b</i>	{8,9,10}	{6,7}	∅	{1,2,3,4,5}
Experiment <i>c</i>	{9,10}	{7,8}	∅	{1,2,3,4,5,6}
Experiment <i>d</i>	{10}	{8,9}	∅	{1,2,3,4,5,6,7}

The objective of these experiments is to find, in each case, the distribution of the four sets considered, in such a way that it shows the possibility of using a query language to establish quality conditions for the recommendations and to discover the most appropriate intervals for the parameter N (number of recommendations provided to each user).

6 Results

Figure 1 contains four graphs, each of which shows the results obtained in each of the experiments using the parameters listed in the previous section (Table 3).

Experiment *a* (Graph 1a) provides the usual values of true-positives which are obtained with *FilmAffinity.com*, where the default relevance threshold is $\theta=9$. The distribution of false-positives in the 3 remaining sets (“relevant”, “slightly relevant” and “not relevant”) is interesting as it highlights the hypothesis of this paper: the quality of the recommendations not only depends on the true-positive values, but rather, it is highly conditioned by the distribution of the false-positives when the range of the set of possible votes is large.

We can observe how the possibility of providing a higher or lower number of recommendations in the RS (*x* axis) is highly influenced by the rating we give to the quality of the set of “relevant” votes. If the rating is that followed by the traditional precision measure, we will only take into account the curve of the “very relevant” set and, therefore, the quality values are sharply reduced as the value of *N* increases. If on the other hand, we sufficiently rate the “relevant” recommendations, the quality values are slightly reduced as the value of *N* increases.

In Experiment *a* we can establish a specific recommendation quality condition, such as:

$$N = (Precision_{very\ relevant} \geq 0.3 \text{ OR } Precision_{relevant} \geq 0.48) \text{ AND } Precision_{slightly\ relevant} \leq 0.2$$

...or fuzzy condition, such as:

$$N = (NotLow(Precision_{very\ relevant}) \text{ OR } Medium(Precision_{relevant})) \text{ AND } Low(Precision_{slightly\ relevant})$$

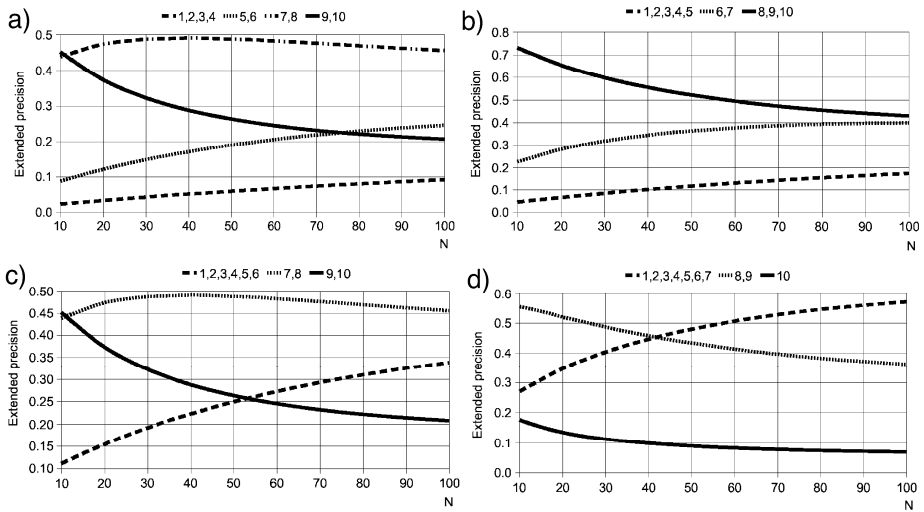


Fig. 1. Results of the experiments *a*, *b*, *c* & *d*. database: *FilmAffinity*, $N \in \{10, \dots, 100\}$, $K=180$, 20% of test users, 20% of test items

Experiment *c* (Graph 1c) is a simplification of experiment *a*, where the set of values “slightly relevant” are transferred to the “not relevant” set. Although we lose flexibility when defining the quality conditions, we gain clarity in the outline, which is now more straightforward: the adequate level of quality is indicated by the “very relevant” set, the inadequate level of quality is indicated by “not relevant” and the modulation of the desired level of quality is essentially established with the “relevant” set.

By establishing the following condition in Experiment *c*:

$$N = Precision_{very\ relevant} \geq 0.35 \text{ OR } Precision_{relevant} \geq 0.48$$

We obtain an approximate range of the number of adequate recommendations:
 $N \in \{10, \dots, 100\} \cup \{5, \dots, 45\}$

In the same way, the query language would return the false value on evaluating the condition:

$$(N=30 \text{ OR } N>50) \text{ AND } (Precision_{very\ relevant} \geq 0.35 \text{ OR } Precision_{relevant} \geq 0.48)$$

Together, Experiments *b* and *d* show the difference in the approaches for the use of the precision and extended-precision quality measures. The first case (Experiment *b*, Graph 1b) establishes a large number of values in the “very relevant” set, and therefore, the RS manager would probably assimilate the appreciation of quality of the RS to that provided by this set, where this is the outline used with the traditional precision measure.

In the case of Experiment *d* (Graph 1d), the “very relevant” set only contains the highest value of the existing rating (the excellent value), making it possible for the “relevant” set to contain values 8 and 9, which are very good values, but not excellent. This way, the “relevant” set can be used as a base to decide the quality conditions or to discover the best values of *N*; we can also make the decision to subdivide the “relevant” set in 2, with the aim of being able to refine the conditions even further.

As an example of the difference in possibilities between experiments *b* and *d*, the following condition provides a satisfactory range of values *N* in Experiment *d*, but not in *b*:

$$\text{select } N \text{ where } Precision_{relevant} > Precision_{not\ relevant}$$

Another example that shows us the advantages of the proposed approach is the use in Experiment *d* of the following condition:

$$N = (Precision_{very\ relevant} \geq 0.1 \text{ OR } Precision_{relevant} \geq 0.4) \text{ AND } Precision_{not\ relevant} < 0.4$$

The final result is determined by the restriction of the last term ($Precision_{not\ relevant} < 0.4$), which provides us with information which we do not have when we use traditional precision as the recommendation quality measure, as in this case the “not relevant” set would be composed of the values 1 to 9.

7 Conclusions and Future Works

RS which admit an extensive set of values in the votes (usually those which admit more than 5 stars to rate an item) cannot be assessed adequately using precision as a recommendation quality measure. This is due to the fact that the division of the possible values of the votes into just two sets, relevant (true-positive) and not-relevant (false-positive), proves to be too poor and involves the accumulation of values in the not-relevant set. In order to establish a balanced quality measure it is necessary to have access to detailed information on how the cases of false-positives are distributed, which are grouped in the imprecise value *I-precision*.

The importance of the extended-precision proposed is determined because the users of the RS are highly sensitive to cases of false-positives, that is, recommendations made which have proved to be not relevant. These situations often

lead to a loss of trust in the system by the users; therefore, every improvement in the recommendation quality measures contributes to providing an increase in satisfaction of the users of the RS.

This paper provides the mathematical formalism which defines the precision quality measure in RS and its generalization to extended-precision, it contributes the results of various experiments on a real RS database with 10 possible votes (*FilmAffinity.com*) and proposes the utilization of a query language, which the administrators of this type of RS will be able to use to improve the results of the recommendations.

This paper could be extended following two independent lines:

1. Introducing fuzzy logic in the definition and processing of the proposed query language.
2. Determining a unified precision measure which weights, according to each RS administrator's criteria, the results obtained with each of the sets established (*very relevant, relevant,...*), in such a way that a Graph can be obtained of the extended-precision of the RS for each value of N considered. A possible approach to define the administrators' criteria would be, once again, fuzzy logic.

Acknowledgments. Our acknowledgement to the *FilmAffinity.com* company.

References

1. Konstan, J.A., Miller, B.N., Riedl, J.: PocketLens: toward a personal recommender system. *ACM Trans. on Inf. Syst.* 22(3), 437–476 (2004)
2. Antonopoulos, N., Salter, J.: Cinema screen recommender agent: combining collaborative and content-based filtering. *IEEE Intell. Syst.*, 35–41 (2006)
3. Li, P., Yamada, S.: A movie recommender system based on inductive learning. In: *IEEE Conf. on Cybern. and Intell. Syst.*, vol. 1, pp. 318–323 (2004)
4. Jinghua, H., Kangning, W., Shaohong, F.: A survey of e-commerce recommender systems. In: *Int. Conf. on Service Syst. and Service Management*, pp. 1–5 (2007)
5. Denis, H.: Managing collaborative learning processes, e-learning applications. In: *29th Int. Conf. on Inf. Technol. Interfaces*, pp. 345–350 (2007)
6. Bobadilla, J., Serradilla, F., Hernando, A.: Collaborative Filtering adapted to Recommender Systems of e-learning. *Knowl. Based Syst.* 22, 261–265 (2009)
7. Lang, K.: NewsWeeder: Learning to filter netnews. In: *12th Int. Conf. on Machine Learning*, Tahoe City, CA (1995)
8. Krulwich, B.: Lifestyle Finder: Intelligent user profiling using large-scale demographic data. *Artificial Intell. Magazine* 18(2), 37–45 (1997)
9. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowl. and Data Eng.* 17(6), 734–749 (2005)
10. Herlocker, J.L., Konstan, J.A., Riedl, J.T., Terveen, L.G.: Evaluating collaborative filtering recommender systems. *ACM Transactions on Inf. Syst.* 22(1), 5–53 (2004)
11. Gao, L.Q., Li, C.: Hybrid personalized recommended model based on genetic algorithm. In: *Int. Conf. on Wireless Commun. Netw. and Mob. Computing*, pp. 9215–9218 (2008)

12. Ho, Y., Fong, S., Yan, Z.: A hybrid ga-based collaborative filtering model for online recommenders. In: *Int. Conf. on e-Business*, pp. 200–203 (2007)
13. Al-Shamri, M.Y., Bharadwaj, K.K.: Fuzzy-genetic approach to recommender Systems based on a novel hybrid user model. *Expert Syst. with Applications* 35, 1386–1399 (2008)
14. Huang, Z., Zeng, D., and Chen, H.: A comparison of collaborative-filtering recommendation algorithms for e-commerce. *IEEE Intelligent Systems*, 68–78 (2007)
15. Sanchez, J.L., Serradilla, F., Martinez, E., Bobadilla, J.: Choice of metrics used in collaborative filtering and their impact on recommender systems. In: *Proceedings of the IEEE International Conference on Digital Ecosystems and Technologies (DEST 2008)*, pp. 432–436 (2008)
16. Bobadilla, J., Serradilla, F., Bernal, J.: A New Collaborative Filtering Metric that Improves the Behavior of Recommender Systems. *Knowl. Based Syst.* 23, 520–528 (2010)
17. Ryan, P.B., Bridge, D.: Collaborative Recommending using Formal Concept Analysis. *Knowl. Based Syst.* 19(5), 309–315 (2006)
18. Leung, C.W., Chan, S.C., Chung, F.L.: An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge Based Systems* 21(7), 515–529 (2008)
19. Hernández, F., Gaudioso, E.: Evaluation of Recommender Systems: a New Approach. *Expert Syst. with Appl.* (35), 790–804 (2008)
20. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: a constant time collaborative filtering algorithm. *Inf. Retr.* 4(2), 133–151 (2001)
21. Breese, J.S., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: *14th Conf. on Uncertain. in Artif. Intell.*, pp. 43–52. Morgan Kaufmann (1998)
22. Candillier, L., Meyer, F., Boullé, M.: Comparing State-of-the-art Collaborative Filtering Systems. In: Perner, P. (ed.) *MLDM 2007. LNCS (LNAI)*, vol. 4571, pp. 548–562. Springer, Heidelberg (2007)
23. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *Adaptive Web 2007. LNCS*, vol. 4321, pp. 291–324. Springer, Heidelberg (2007)

A Tool for Link-Based Web Page Classification^{*}

Inma Hernández, Carlos R. Rivero, David Ruiz, and Rafael Corchuelo

University of Seville
Seville, Spain

{inmahernandez, carlosrivero, druiiz, corchu}@us.es

Abstract. Virtual integration systems require a crawler to navigate through web sites automatically, looking for relevant information. This process is online, so whilst the system is looking for the required information, the user is waiting for a response. Therefore, downloading a minimum number of irrelevant pages is mandatory to improve the crawler efficiency. Most crawlers need to download a page to determine its relevance, which results in a high number of irrelevant pages downloaded. In this paper, we propose a classifier that helps crawlers to efficiently navigate through web sites. This classifier is able to determine if a web page is relevant by analysing exclusively its URL, minimising the number of irrelevant pages downloaded, improving crawling efficiency and reducing used bandwidth, making it suitable for virtual integration systems.

Keywords: Crawling, Web Page Classification, Virtual Integration.

1 Introduction

Virtual Integration aims at accessing web information in an automated manner, retrieving information relevant to a user query from the Web. Automated access to the Web requires a crawler, which is a tool able to navigate through web sites automatically, looking for relevant information. Traditional crawlers visit every link on every page, download their target, and check whether the page contains relevant information. This means that, even when a page is irrelevant, the crawler has to download it and check if it is relevant or not, which results in a large number of irrelevant pages downloaded.

Note that the Virtual Integration process is online, which means that whilst the system is looking for the required information, the user is waiting for a response. Therefore, downloading a minimum number of irrelevant pages is mandatory to improve the crawler efficiency, which is a concern for several researchers [\[9,15,26\]](#).

There are some techniques that improve traditional crawlers efficiency by endowing the crawler with classification skills. For example, focused crawlers

^{*} Supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2008-04718-E, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, and TIN2010-09988-E).

find pages belonging to one or more topics exclusively, so they are supported by a content-based classifier that determines whether each page belongs to those topics [11,14,22,24]. Other crawlers include classifiers based on other features like page structure [19,20,27]. Finally, there are crawling techniques that rely completely on the user to define navigation patterns [2,5,8,23,29].

In this paper, we focus on web sites that follow a certain navigational pattern, which is the most common pattern in the Web [19]. This pattern starts with a form page; then, after users submit a query, the system returns a hub, i.e., a page containing an indexed list of answers to it, each of which contains just a brief description and a link to a detail page. Note that the term “hub” is based on the hub and authority concepts introduced by Kleinberg [18].

In this kind of web sites, hubs are created by instantiating scripts with data stored in a database [7]. This means that all hubs from the same web site share a common template, usually in the form of headers, footers and side bars containing navigational aids, copyright information and advertising [30], which frame the page areas that contain the information that varies from hub to hub. Similarly, URLs that point to each hub and detail page are generated as well by the same process of filling a URL pattern with keywords that identify the generated page. Therefore, all URLs from a certain site can be expressed by a collection of URL patterns.

We propose a classifier that helps crawlers to efficiently navigate through web sites, by determining if a web page is relevant by analysing exclusively its URL. Our classifier is different to existing proposals, since it is based on features that are not in the page to be classified, but in pages that link to it. Therefore, it is not necessary to download a page to classify it, which avoids downloading irrelevant pages, reducing the bandwidth and making it efficient and suitable for Virtual Integration systems. Moreover, our proposal is automated, requiring a minimum intervention from users. Furthermore, our classifier is trained using an unlabelled training set of URLs, thus relieving the user from the tedious task of assigning a label to each training page.

Our hypothesis is that there is usually a correspondence between URL patterns and the concept contained in the pages with URLs following that pattern, so that we can classify web pages containing different concepts by means of the pattern matching their URL. Therefore, our classification technique consists on finding the different URL patterns or prototypes that compose links in a given web site. Then, we use these prototypes to classify links by template matching. Furthermore, our technique is able as well to detect links belonging to the Web site template.

The rest of the paper is structured as follows. Section 2 presents the related work in the web page classification area; Sections 3 and 4 introduce the core definitions that will be used throughout the paper; Section 5 describes the tool design; Section 6 presents the evaluation of our tool; finally, Section 7 lists some of the conclusions drawn from the research and concludes the paper.

2 Related Work

Web page classification has been extensively researched, and several techniques have been applied with successful experimental results. In general, we catalogue classifiers according to the type and location of the classification features. There are three main trends in feature types: content-based, structure-based and hybrid classifiers. As for feature location, most approaches obtain features from the page to be classified, whilst others get them from neighbour pages.

Content-based classifiers ([17,25]) categorize a web page according to the words and sentences it contains. These kinds of classifiers group all pages within the same topic, assigning them the same class label. As for structure-based classifiers ([3,4,6,13,27,28]), the main feature used to classify pages is their physical organisation of contents, usually expressed in a tree-like data structure, like a DOM Tree. Also, there are hybrid approaches [10,21] which take both content and structural features into account.

All previous classifiers consider different kinds of features, but in most cases those features are extracted from the page to be classified, which requires downloading it previously. There are also classifiers that explore the possibility of classifying a web page by using features extracted from neighbour pages, instead of the page itself, being the neighbour of a page another page that has a link to the former, or, conversely, that is linked from it. All these proposals are content-based, and usually rely on features such as the link anchor text, the paragraph text surrounding the anchor [12], the headers preceding the anchor, the words in the URL address, or even a combination of them [16]. If the link is surrounded by a descriptive paragraph or the link itself contains descriptive words, it is possible to decide the page topic in advance of downloading it.

3 Core Definitions

In this Section, we introduce some preliminary concepts that will be used throughout the rest of the paper.

Hub. Each hub is defined by the set of links that it comprises, $H_i = \{l_1, l_2, l_3, \dots, l_m\}$

Hubset. Set of hubs obtained from a particular site. $H = \{H_1, H_2, H_3, \dots, H_n\}$

Linkset. Set of links that are comprised in a hubset H , $L = \bigcup_{i=1}^n H_i \in H$

Link. Tuple that represents a URL, $l = (S, A, P, N, V)$. Links are obtained from URLs by means of a tokeniser, according to RFC 3986, where S is the schema of the URL, A is its authority or domain name, P is a sequence of path segments, N is a sequence of names of the parameters in the URL query string and V is the sequence of the former parameters values. For the sake of simplicity, throughout the paper we use the notation X to refer to any of the sequences P , N or V .

Prototype. Link $p = (S, A, P, N, V)$ that represents a URL pattern, where each element in P , N and V is either a literal or a wildcard, \star . A wildcard represents any sequence of characters (excluding separators '?', '/', '#', '=' and '&')

Common Path Links. Let L be a linkset from a given site, l be a link in L and $X(i)$ be the i -th element of sequence X in l . We define the set $CPL_X(l, i)$ as the set of all links l' in L having the same prefix as l up to (and excluding) $X(i)$. Recall that a prototype is a link that includes some wildcard sections, so we can calculate the CPL set of a prototype likewise.

4 Classification Features

In this Section, we introduce the features that support building the set of prototypes that represent all links in a given site. We take a statistical approach to the problem of prototype building, and we base our technique in the definition of probabilistic features for each link and each token inside a link. First we give a formal definition of these features and later we illustrate their use by means of an example.

4.1 Features Definition

Definition 1 (Link feature). Let H be a hubset from a certain web site with size n , and l be a link $l \in H$. Probability F_L of a link in the context of H is defined as follows.

$$F_L(l) = \frac{|\{H_i \in H \cdot l \in H_i, i \in [1, n]\}|}{n} \tag{1}$$

In Equation 1, we must assure that the hubset is sufficiently large so that the probability estimation is statistically significant, hence we require that $|H| \geq 30$, which is the usual threshold in statistical literature. $F_L(l)$ takes values in the range $[1/n, 1]$. Links that appear more frequently in hubs from a hubset, have a higher F_L than those appearing just in a few of them, to the point that links with $F_L = 1$ appear in every single $H_i \in H$. At the other end of the distribution, links with F_L near to 0 never appear in any of H hubs.

As an example, Figure 1a shows the histogram of F_L values obtained from 100 hubs in an e-commerce site (Amazon.com) an two academic sites (Microsoft Academic Search and TDG Scholar).

Definition 2 (Tokens Features). Let H be a hubset from a given site with size n , L its linkset, l a link of the form (S, A, P, N, V, Q) in L and $X(i)$ be the i -th element of X , we define the feature value of $X(i)$ given as the following probability.

$$F_X(l, i) = \frac{|\{H_j \in H \cdot H_j \cap CPL_X(l, i + 1) \neq \emptyset, j \in [1, n]\}|}{n} \tag{2}$$

These features values are in the same range as F_L , $[1/n, 1]$. Same as with F_L , path segments that appear more frequently in hubs from H have a higher F_P than those that only appears in URLs from some of the hubs.

Figure 1b shows the histogram of F_P , F_N and F_V values from the same hubsets and sites as defined for F_L values. It is noticeably similar to the F_L histogram presented earlier, with the majority of values around $1/n$, and just a small tail near 1.

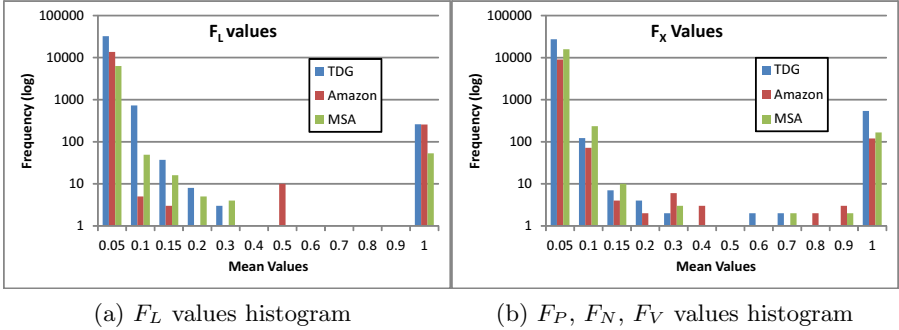


Fig. 1. F_P , F_N , F_V and F_L values histogram, from sites: Amazon.com, TDG Scholar and Microsoft Academic Search

Given that a prototype has the same signature as a link, both previous definitions 1 and 2 are applicable as well to prototypes. For the sake of simplicity, we assume that $F_P(p, i) = 1$ iff $p = \{S, A, P, N, V, Q\} \wedge P(i) = \star$ (similarly, with F_N and F_V).

4.2 Features Examples

Example 1. Consider an experiment over Amazon.com, in which we issue 100 queries using the top 100 words in English language, discarding stop words. The result of this experiment is a hubset H composed of $n = 100$ hubs. The F_L values calculated for some of the links in H are shown in Table 1.

All Amazon pages contain a navigation bar in their upper part, including links such as “Home” “Sign In” and “Help”. Examples of these links URLs are, respectively, links with ID 2, 3 and 4, and they are always present in every page from the site. Therefore, for any hubset extracted from Amazon, the probability of these URLs is always 1.

On the other side, there are links whose appearance depends on the specific page being considered. For example, links to a page with detailed information about a product, just like links with ID 1, 5 and 6 in the example, only appear in hubs which are answers to certain queries. Therefore, its probability depends on the hubset, although we can assume that, for a random set of hubs, F_L value is rather low.

In general, our hypothesis is that for links whose F_L in a hubset H is not 1 (or near 1), it is in fact around $1/n$, i.e., probability values are grouped around the two extremes of the distribution (0 and 1), and the number of links whose probability is in the middle of the distribution is very low. Back to Figure 1a, we observe that most values are grouped around 0.05, which means that most links just appear in a range of 1 to 5 hubs, approximately. We must note that there is a small but significant group of values around 1, i.e., the group of links that are present in every hub from the site. We can therefore conclude that links with $F_L = 1$ are those belonging to the site template. Hence, our technique allows us to detect the template of a given site, besides classifying its links according the concept contained in their targets.

Table 1. Values for feature F_L in Example 1

ID	l	$F_L(l)$
1	http://www.amazon.com/Head-First-Java/dp?ie=UTF8&qid=130	0.01
2	http://www.amazon.com/ref-gno_logo	0.99
3	http://www.amazon.com/Help/b/ref-topnav_help?ie=UTF8&node=508510	0.99
4	http://www.amazon.com/gp/yourstore/ref-pd_irl_gw?ie=UTF8&signln=1	1.00
5	http://www.amazon.com/Effective-Java/dp?ie=UTF8&qid=130	0.01
6	http://www.amazon.com/Head-First-Java/product-reviews?ie=UTF8	0.03

Let l_1 be the link with ID = 1 in previously defined H . After the experiment, we obtain the values for features F_P , F_N and F_V presented in Table 2a. As a comparison, in Table 2b, we show the values for features F_P , F_N and F_V for the prototype p that results when we replace the first path segment in l_1 (“Head-First-Java”) with a wildcard.

Table 2. Values for features F_P , F_N and F_V for l_1 and p , in Example 1

	X(i)	Value		X(i)	Value
$F_P(l_1, 1)$	Head-First-Java	0.01	$F_P(p, 1)$	★	1
$F_P(l_1, 2)$	dp	0.01	$F_P(p, 2)$	dp	0.98
$F_N(l_1, 1)$	ie	0.01	$F_N(p, 1)$	ie	0.99
$F_N(l_1, 2)$	qid	0.01	$F_N(p, 2)$	qid	0.99
$F_V(l_1, 1)$	UTF-8	0.01	$F_V(p, 1)$	UTF-8	0.99
$F_V(l_1, 2)$	123	0.01	$F_V(p, 2)$	123	0.01

(a) Values for l_1

(b) Values for p

Based on the former example, we can extract some conclusions from the different values of F_P , F_N and F_V . For example, token “dp”, with $F_P(p, 2) = 0.98$, is a fixed part of every link to Amazon product detail pages, and therefore, it is more frequent throughout the site than token “123”, whose $F_V(p, 2)$ is near 0 as it is a parameter that identifies queries, and therefore, it is different for every issued query. As a result, its F_V value is 0.01, indicating that it just appears in links from a single hub. Similarly, parameter 1, with name “ie” and value “UTF-8”, is also a fixed part in all Amazon links, so their F_N and F_V values respectively are near to 1 in Table 2b.

Our hypothesis regarding F_P , F_N and F_V values is the same exposed earlier for F_L values. In this case, the straightforward application is to build prototypes: tokens with a near-zero value are not relevant, so we can abstract over them and obtain a more general representation of all such segments in the form of a regular expression, i.e., of a prototyping token. Meanwhile, tokens with a feature value significantly higher than the others (usually around 1) appear in most hubs, so they are part of the characteristic URL patterns used to compose site URLs, i.e., they are relevant, so we keep them as literals.

5 Classification Tool

Based on the previous features, we implemented a link classifier, following the architecture in Figure 2. First, a training set is needed, composed by links from

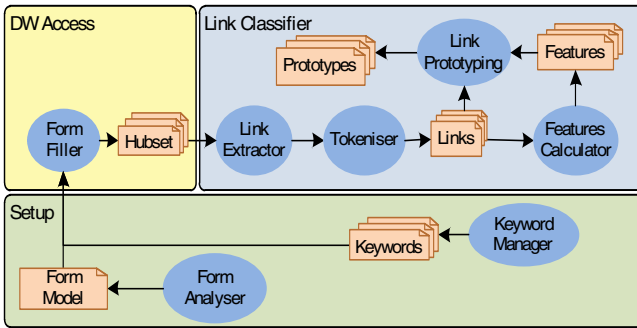


Fig. 2. Workflow of the architecture

the site we wish to extract information from. For this purpose, we make use of the Form Analyser which analyses the forms to obtain a form model, and the Form Filler that uses this model to automatically fill in the form and retrieve the resulting hubs, composing a hubset. Our proposal is focused on keyword-based queries, hence the form filler only deals with forms that contain at least one text field. A Keyword Manager is responsible for finding a corpus of keywords to be used by the form filler, trying to obtain the maximum number of hubs as possible, minimising the keywords that yield no result.

Afterwards, all URLs from the retrieved hubset are extracted and tokenised. For each link, values of features F_P , F_N , F_V and F_L , as defined in section 4 are calculated, and used to build an ordered set of prototypes, where each prototype represents a different class of links, i.e., links leading to pages containing a different concept. Some prototypes may subsume other prototypes, i.e., a regular expression that is more general than other, and that matches all links matched as well by the latter. To avoid misclassifications, in cases like that we always give a higher priority to the most specific prototype.

6 Evaluation

We developed a proof-of-concept application, based on the former architecture. An example of the classification results is presented in Figure 3. We observe that Cluster 0 represents the site template links, Cluster 8 products (<http://www.amazon.com/★/dp/★>), Cluster 9 product reviews (<http://www.amazon.com/★/product-reviews/★>) and Cluster 12 authors (<http://www.amazon.com/★/e/★>), amongst others.



Fig. 3. Example of Link Classification: Amazon.com hub page

We performed an experiment to test our tool, evaluating the most relevant concepts on three different sites. The classification was evaluated by means of 10-fold cross evaluation, obtaining values for precision, recall and f1-measure in Table 3. For each measure, we show its mean value, as well as the confidence interval of 95%.

Table 3. Evaluation results

Site	Concept	Precision	Recall	F1-Measure
Amazon	Products	0.978 ± 0.022	0.703 ± 0.033	0.818 ± 0.011
	Reviews	0.978 ± 0.029	0.705 ± 0.031	0.819 ± 0.030
TDG Scholar	Authors	0.908 ± 0.005	0.761 ± 0.004	0.828 ± 0.014
Ms Academic	Papers	0.979 ± 0.003	0.864 ± 0.006	0.851 ± 0.023

We observe that recall values are always lower than precision. We have concluded that our proposal yields prototypes that are too specific, so our future work is focused on improving these results by means of post processing.

7 Conclusions

Our proposal classifies pages according to their URL format without downloading them beforehand, saving bandwidth and time. Parting from an unlabelled set of links, a set of prototypes is built, each of which represents all links to pages containing a concept embodied in a particular web site. The resulting prototype set can be used by a crawler to improve its efficiency by selecting in each page only links leading to pages with concepts that are interesting for the user, reaching those pages whilst downloading the minimum number of irrelevant pages. Besides, our classifier is able to detect the template of a web site, i.e., links that appear in every page in the site, and hence will most probably not lead to information related to that query.

There are some proposals that classify pages according to the text surrounding the link in the referring page. This is not a general technique, given that not all links include in their surroundings words useful for classification. Our proposal classifies web pages depending on the link URL format, so it is not only efficient, but also generic and applicable in different domains. Besides, user supervision is kept to a minimum, given that the classifier is trained using an unlabelled set of links collected automatically.

References

1. Aggarwal, C.C., Al-Garawi, F., Yu, P.S.: On the design of a learning crawler for topical resource discovery. *ACM Trans. Inf. Syst.* 19(3), 286–309 (2001)
2. Anupam, V., Freire, J., Kumar, B., Lieuwen, D.F.: Automating web navigation with the webcr. *Comp. Netw.* 33(1-6), 503–517 (2000)
3. Arasu, A., Garcia-Molina, H.: Extracting structured data from web pages. In: *SIGMOD*, pp. 337–348 (2003)
4. Bar-Yossef, Z., Rajagopalan, S.: Template detection via data mining and its applications. In: *WWW*, pp. 580–591 (2002)
5. Bertoli, C., Crescenzi, V., Merialdo, P.: Crawling programs for wrapper-based applications. In: *IRI*, pp. 160–165 (2008)
6. Blanco, L., Crescenzi, V., Merialdo, P.: Structure and semantics of Data-IntensiveWeb pages: An experimental study on their relationships. *J. UCS* 14(11), 1877–1892 (2008)
7. Blanco, L., Dalvi, N., Machanavajjhala, A.: Highly efficient algorithms for structural clustering of large websites. In: *WWW 2011*, pp. 437–446. *ACM* (2011)
8. Blythe, J., Kapoor, D., Knoblock, C.A., Lerman, K., Minton, S.: Information integration for the masses. *J. UCS* 14(11), 1811–1837 (2008)
9. Boldi, P., Codenotti, B., Santini, M., Vigna, S.: UbiCrawler: a scalable fully distributed web crawler. *Softw., Pract. Exper.* 34(8), 711–726 (2004)
10. Caverlee, J., Liu, L.: Qa-pagelet: Data preparation techniques for large-scale data analysis of the deep web. *IEEE Trans. Knowl. Data Eng.* 17(9), 1247–1262 (2005)

11. Chakrabarti, S.: Focused web crawling. In: *Encyclopedia of Database Systems*, pp. 1147–1155 (2009)
12. Cohen, W.W.: Improving a page classifier with anchor extraction and link analysis. In: *NIPS*, pp. 1481–1488 (2002)
13. Crescenzi, V., Mecca, G., Merialdo, P.: RoadRunner: Towards automatic data extraction from large web sites. In: *VLDB*, pp. 109–118 (2001)
14. de Assis, G.T., Laender, A.H.F., Gonçalves, M.A., da Silva, A.S.: Exploiting Genre in Focused Crawling. In: Ziviani, N., Baeza-Yates, R. (eds.) *SPIRE 2007*. LNCS, vol. 4726, pp. 62–73. Springer, Heidelberg (2007)
15. Edwards, J., McCurley, K.S., Tomlin, J.A.: An adaptive model for optimizing performance of an incremental web crawler. In: *WWW*, pp. 106–113 (2001)
16. Fürnkranz, J.: Hyperlink ensembles: a case study in hypertext classification. *Inf. Fusion* 3(4), 299–312 (2002)
17. Hotho, A., Maedche, A., Staab, S.: Ontology-based text document clustering. In: *KI*, vol. 16(4), pp. 48–54 (2002)
18. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *J. ACM* 46(5), 604–632 (1999)
19. Lage, J.P., da Silva, A.S., Golgher, P.B., Laender, A.H.F.: Automatic generation of agents for collecting hidden web pages for data extraction. *Data Knowl. Eng.* 49(2), 177–196 (2004)
20. Liddle, S.W., Embley, D.W., Scott, D.T., Yau, S.H.: Extracting Data Behind Web Forms. In: Olivé, À., Yoshikawa, M., Yu, E.S.K. (eds.) *ER 2003*. LNCS, vol. 2784, pp. 402–413. Springer, Heidelberg (2003)
21. Markov, A., Last, M., Kandel, A.: The hybrid representation model for web document classification. *Int. J. Intell. Syst.* 23(6), 654–679 (2008)
22. Mukherjea, S.: Discovering and analyzing world wide web collections. *Knowl. Inf. Syst.* 6(2), 230–241 (2004)
23. Pan, A., Raposo, J., Álvarez, M., Hidalgo, J., Viña, Á.: Semi-automatic wrapper generation for commercial web sources. In: *EISIC*, pp. 265–283 (2002)
24. Pant, G., Srinivasan, P.: Link contexts in classifier-guided topical crawlers. *IEEE Trans. Knowl. Data Eng.* 18(1), 107–122 (2006)
25. Selamat, A., Omatu, S.: Web page feature selection and classification using neural networks. *Inf. Sci.* 158, 69–88 (2004)
26. Shkapenyuk, V., Suel, T.: Design and implementation of a high-performance distributed web crawler. In: *ICDE*, pp. 357–368 (2002)
27. Vidal, M.L.A., da Silva, A.S., de Moura, E.S., Cavalcanti, J.M.B.: Structure-based crawling in the hidden web. *J. UCS* 14(11), 1857–1876 (2008)
28. Vieira, K., da Silva, A.S., Pinto, N., de Moura, E.S., Cavalcanti, J.M.B., Freire, J.: A fast and robust method for web page template detection and removal. In: *CIKM*, pp. 258–267 (2006)
29. Wang, Y., Hornung, T.: Deep web navigation by example. In: *BIS (Workshops)*, pp. 131–140 (2008)
30. Yi, L., Liu, B., Li, X.: Eliminating noisy information in web pages for data mining. In: *KDD*, pp. 296–305 (2003)

Information Extraction for Standardization of Tourism Products

Nuno Miranda¹, Ricardo Raminhos¹, Pedro Seabra¹
Teresa Gonçalves², José Saias², and Paulo Quaresma²

¹ VIATECLA SA, Almada, Portugal

{nmiranda, rraminhos, pseabra}@viatecla.com

² Dep. Informática, Universidade de Évora, Évora, Portugal
{tcg, jsaias, pq}@di.uevora.pt

Abstract. Tourism product descriptions are strongly supported on natural language expressions. Appropriate offer selection, according to tourist needs, depends highly on how these are communicated. Since no human interaction is available while presenting tourism products online, the way these are presented, even when using only textual information, is a key success factor for tourism web sites to achieve a purchase. Due to the large amount of tourism offers and the high dynamics in this sector, manual data management is not a reliable or a scalable solution. This paper presents a prototype developed for automatic extraction of relevant knowledge from tourism-related natural language texts. Captured knowledge is represented in a normalized format and new textual descriptions are produced according to available marketing channels. At this phase, the prototype is focused on hotel descriptions and is already using real operational data retrieved from the KEYforTravel tourism platform.

1 Introduction

Online presence of tourism related web sites has accompanied the exponential growth of the Internet. As an example, US online tourism revenues between 2001–2004 have increased on average 29% per year [5]. An important subset of these revenues, correspond to commercial web sites (e.g. *Expedia*) where users can browse and search tourism offers, inspect details, perform and fulfill reservations by themselves, following a self-booking tool approach.

Even today, when multimedia is increasing its presence on the web, natural language textual descriptions still remain by far the mostly used format for e-marketing and promotion applied to tourism products (i.e. hotel, aviation, rent-a-car, holiday packages). Descriptions presented to the user are provided by external service connectors (e.g. *GTA* or *HotelBeds* for hotel products) or updated manually by the tourism online operator. Offers are structured differently (complementing or overlapping each other) and mostly consists on simple enumerations of available services and equipments. Usually these textual descriptions are presented to the end user as obtained from the service provider without prior preparation.

This work presents a prototype developed for the automatic extraction of relevant knowledge from tourism-related textual expressions that enables the creation of appropriate descriptions according to user profile in order to provide a suitable segmented e-marketing and promotion process. For now, the prototype is focused on the hotel descriptions subset using real operational data retrieved from the KEYforTravel [22] tourism platform.

The paper is organized as follows: Section 2 presents the application domain and Section 3 introduces the Information Extraction and Text Classification tasks. Section 4 describes the System Architecture and Experiments and Evaluation are carried out on Section 5. Finally, Section 6 presents some conclusions and points out possible future work.

2 The Application Domain

Hotel descriptions are commonly available in natural language text. It usually contains information about:

- hotel services commonly shared by all tourists,
- equipment made available on each room and
- location information, normally relative to some well-known point of interest (e.g. street, monument or metro station).

Each description tries to summarize (in the minimum amount of text) the multiple features and benefits made available by the hotel. Together with price and availability factors, they are responsible for attracting customers.

Although all relevant information is comprised within the description, it is mostly used for presentation purposes. In this way, all individual knowledge isn't potentiated for searching and offer refinement purposes (e.g. search for all hotels with jacuzzi and swimming pool located nearby a metro station in London).

Hotels requiring strong market projection are usually present in one or more hotel service aggregators. Commercial tourism web sites can use multiple of these services to present hotel offers world-wide, resulting in possible multiple descriptions for the same hotel. Depending on each connector business focus, some hotel features may be found more or less relevant, resulting in disjoint descriptions. When detected, one of these descriptions is usually elected as primary and becomes the only one to be considered for hotel presentation (all others descriptions are discarded, as well as their complementary information, even if not present in the primary description). Further, for commercial tourism sites that provide world-wide hotel offer (in the order of several thousands) it is not possible to individually manage each hotel description. Thus, many commercial sites choose to present directly to the user the description made available by the hotel connector. This results in three main problems:

- there is no differentiation between tourism online operators sharing the same connectors
- descriptions are not targeted to the user/market segmentation

- descriptions are not controlled nor normalized. This results on different, heterogeneous descriptions presented altogether.

Building a system that extracts all relevant hotel features and normalizes them it is possible to address the previous posed problems.

Hotel descriptions are stored in the KEYforTravel platform, that gathers them from several external service connectors (like *GTA* or *HotelBeds*). Thus, the system goal is to normalize information from different sources and aggregate it in KEYforTravel clients in a standardized way (both in a structured way like tables or a natural language one).

3 Information Extraction and Text Classification

This section introduces the information extraction task and the text classification problem along with the all-purpose classification algorithms used in this work.

3.1 Information Extraction

Information Extraction is a type of information retrieval whose goal is to automatically extract structured information (categorized, contextually and semantically well-defined data) from a certain domain, from unstructured machine-readable documents. This has been an active area of research, exhibited in a series of Message Understanding Conferences (MUCs) [6,23] and more recently in ACE evaluations [12].

Detecting entities in natural language text typically involves disambiguating phrases based on the words in the phrase, and the text surrounding the candidate entity. Explored approaches include hand-crafted rules [7], rule learners [1] and other machine learning approaches (e.g. [2]). Another line of research generates probabilistic models. Hidden Markov Models (HMMs) are popular sequential models that have been used in the context of IE [4], as well as other frameworks like Conditional Markov Models [9] and Conditional Random Fields [10].

3.2 Text Classification

Text Classification is a well studied task with many effective techniques. Nowadays, the most popular and successful algorithms for text classification are based on machine learning techniques. Several algorithms have been applied, such as decision trees [20], linear discriminant analysis and logistic regression [18], Naïve Bayes classifier [14] and Support Vector Machines (SVM) [8].

Learning systems have the advantage of flexibility since the only required human effort is to provide a consistent set of labeled examples. Originally, research in text classification addressed the binary problem, where a document is either relevant or not w.r.t. a given category. In real-world situation, however, the great variety of different sources and hence categories usually poses multi-class classification problem, where a document belongs to exactly one category selected from

a predefined set. Even more general is the case of multi-label problem, where a document can be classified into more than one category. This kind of classification problem is typically solved by dividing it into a set of binary classification problems, where each concept is considered independently.

Documents must be pre-processed to obtain a structured representation to be fed to the learning algorithm. The most common approach is to use the bag-of-words representation [17], where each document is represented by the words it contains (their order and punctuation are ignored). Normally, words are weighted by some measure of word's frequency in the document and, possibly, the corpus. In most cases, a subset of words (stop-words) is not considered, because they do not have discriminating power over different classes; some works reduce semantically related terms to the same root applying a lemmatizer.

Decision Tree. A text classifier represented by a decision tree is a tree in which internal nodes represent words, branches represent values which the words may have and the leaves represent classes. It classifies a document verifying recursively (from the root), the node's word and traversing the branch with the document word's value until reaching a leaf; the document belongs to the class labeled by the leaf. The induction of a decision tree is achieved applying a divide and conquer strategy: it verifies if all examples have the same label and, if not, selects a word w and divides the document set on subsets with the same value for w , putting each subset into distinct subtrees. This process is repeated recursively for each of the subtrees until all leaves contain only instances of the same class, which is then chosen as the document label. The key step of the algorithm is the choice of the word w on which the partition is made. This choice is usually made according to a criterion of mutual information or entropy.

Naïve Bayes Classifier. Naïve Bayes classifier is a probabilistic classifier that sets the class of a given document by choosing the class that maximizes the probability of the document, given its attributes. This probability is calculated applying the Bayes theorem and assuming that each attribute is independent from the others. Even naively assuming attribute independence, this algorithm has shown good results on text classification.

Support Vector Machines. Support Vector Machines was motivated by theoretical results from the statistical learning theory: it joins a kernel technique with the structural risk minimization framework. *Kernel techniques* comprise two parts: a module that performs a mapping from the original data space into a suitable feature space and a learning algorithm designed to discover linear patterns in the (new) feature space. The *kernel function*, that implicitly performs the mapping, depends on the specific data type and domain knowledge of the particular data source. The *learning algorithm* is general purpose and robust. It's also efficient, since the amount of computational resources required is polynomial with the size and number of data items, even when the dimension of the embedding space (the feature space) grows exponentially [19]. A mapping example is illustrated in Fig. 1(a).

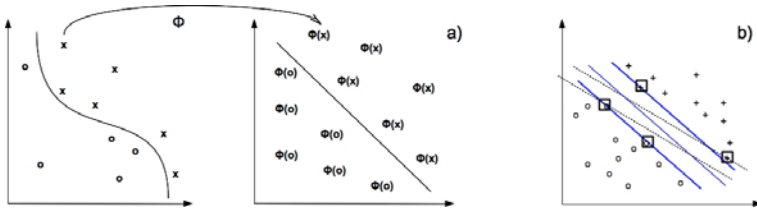


Fig. 1. The SVM approach: kernel transformation and search for maximum margin

The *structural risk minimization* (SRM) framework creates a model with a minimized VC (Vapnik-Chervonenkis) dimension [21]. This theory shows that when the VC dimension is low, the expected probability of error is also low, which means good performance on unseen data (good generalization). In geometric terms, it can be seen as a search to find, between all decision surfaces that separate positive from negative examples, the one with maximum margin (the one having a separating property that is invariant to the most wide translation of the surface). This property is enlighten in Fig. 1b) for a 2-dimensional problem.

4 System Architecture

The system aims to receive an hotel description and produce a standardized version of it. It was designed using a divide and conquer strategy where several small tools that focus on specific and simpler problems were interconnected. There are four main tools: a *Sentence Classifier*, an *Entity Extractor*, an *Ontology Instantiator* and an *Ontological Translator*, that where packed into a Web Service for the system to be available online. This architecture is depicted on Fig. 2, with the information flow between tools also represented.

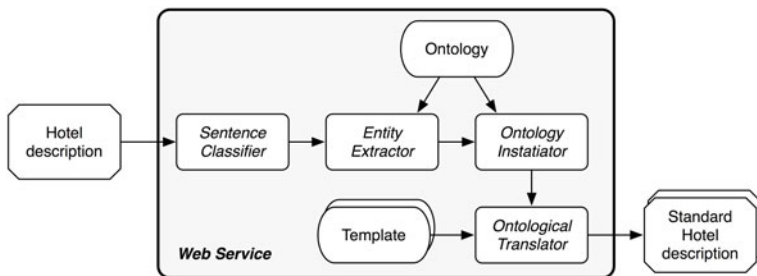


Fig. 2. System architecture diagram

4.1 Sentence Classifier

This prototype's module is responsible for examining and classifying chunks of natural language text. It receives the crude description of the hotel and divides it into sentences. The sentences are then individually examined and automatically classified into a set of predefined classes such as **Equipment**, **Service** and **Location**. Each sentence can belong to more than one class, provided it contains elements of those classes, or none of them, if it doesn't present any evidence. This classification aims at filtering the sentences by type to ensure a specific treatment to each one by the *Entity Extractor* module.

The *Sentence Classifier* was built using machine learning approach and because sentences can belong to more than one class, we are in presence of a multi-label text classification problem. After yielding the set of experiments (section 5.1) the classifiers were built using a Naïve Bayes classifier with the bag-of-words representation of the sentences with binary word weights.

The classifiers were built with WEKA [25], a software package developed by New Zealand Waikato University that implements a collection of ML algorithms.

4.2 Entity Extractor

Having the sentences classified and grouped by type, the system tries to extract useful entities. This is the goal of this module that comprises two steps: finding useful entities and dealing with misspelled words.

Due to the fact that hotel descriptions are given in natural language without any pattern or consistency, the same entity can be described not only by a single term but by a set of synonyms, or even by abbreviations. It can also be the case where the entity reference is misspelled or have failures in diacritics. This last hypothesis is very common since raw descriptions are frequently translated to different idioms and lose the regional diacritics.

To find useful entities on the description of each type of sentence a pattern matching approach was used. This was accomplished by defining a set of some regular expressions able to identify synonyms, abbreviations and "almost" well-written words (e.g. **television**, **T.V.** and **TV** or **mini-bar** and **mini bar**) for common terms used for describing that kind of information.

To cope with misspelled words, the similarity between words not yet extracted and the ones considered relevant to the sentence's type is measured using the Levenshtein distance [11].

4.3 Ontology Instantiator

To retain the entities extracted from the texts, and aiming to provide the basic structure and organization of the involved concepts, an ontology for hotel domain was developed. This ontology comprises 89 classes e 88 relations.

Although in the present architecture ontology instances are the input for the *Ontology Translator*, this normalized knowledge (easily computable) can be applied to substantially expand and improve search capabilities in tourism offers

since each **Service**, **Equipment** or **Location** item can be used in the query itself, or as a parameter in the search results refinement process. Further, since knowledge is formalized using an hierarchical structure, it may be applied to graphically map related items as well as structure navigation.

The hotel ontology was conceived using the Web Ontology Language [24] (OWL), a language designed to be used by applications that need to process Web information content. It facilitates machine interpretability by providing additional vocabulary along with a formal semantics, that besides defining structure, considers possible semantic relationships between objects and attributes.

Each ontology object contains the set of regular expressions and Levenshtein functions used by the *Entity Extractor* module. In this way, the *Entity Extractor* becomes independent of the specific problem at hand.

Using the ontology, this module generates an OWL instance populated with the extracted entities jointly with their attributes and semantic relationships. This instance is then accessed using the Jena Semantic Web Framework [3].

4.4 Ontology Translator

This module is responsible for turning the extracted information attractive and easy to read by humans giving it different flavors according to the preference of the tour operator or the target audience (e.g. corporate versus leisure clients).

This module uses a XML template giving the skeleton for the final information representation and filling it with the extracted information.

5 Experiments and Evaluation

This section presents the experiments done to build the *Sentence Classifier* and the outputs generated by the system when presented with an hotel description.

5.1 Experimental Setup

To build the *Sentence Classifier* we made several experiments with different bag-of-words term weighting representations and different classification algorithms:

- binary, word count and tfidf [17] normalized to unit length term weighting;
- decision tree [16], naïve Bayes [13] and support vector machine (SMO, sequential minimal optimization [15]) algorithms;

The algorithms were run with their default parameters and the model was evaluated using a 10-fold stratified cross-validation procedure with 95% confidence level significance tests.

5.2 Results

We are interested achieving maximum recall, at the cost of lower precision, because false positives are treated later by the *Entity Extractor* of that class. Nevertheless, Table 1 shows recall, precision and F_1 measures for each class, term weight and algorithm.

From the results we can see that the highest recall is achieved by the naïve Bayes classifier with the binary term weight. Bold-face values are significantly better than that setting while italic-face are significantly worse. Also, F_1 values for that setting are only significantly worse than SVM algorithm for the Equipment class.

Table 1. Precision, recall and F_1 values for each category

		Services			Equipment			Location		
term weight	classifier	rec	prec	F_1	rec	prec	F_1	rec	prec	F_1
binary	nBayes	.987	.795	.876	.969	.657	.776	.919	.703	.792
	SVM	<i>.816</i>	.912	.854	<i>.843</i>	.951	.883	<i>.724</i>	.893	.789
	dTree	<i>.601</i>	.863	<i>.696</i>	<i>.724</i>	.961	.802	<i>.589</i>	.955	.713
word	nBayes	<i>.844</i>	.952	.890	<i>.693</i>	.990	.800	<i>.510</i>	.988	<i>.658</i>
	SVM	<i>.830</i>	.905	.858	<i>.858</i>	.951	.893	<i>.711</i>	.892	.779
	dTree	<i>.608</i>	.858	<i>.699</i>	<i>.689</i>	.954	.775	<i>.623</i>	.956	.736
tfidf	nBayes	<i>.868</i>	.969	.911	<i>.749</i>	.974	.835	<i>.451</i>	.960	<i>.592</i>
	SVM	<i>.846</i>	.904	.866	<i>.865</i>	.942	.893	<i>.700</i>	.891	.771
	dTree	<i>.742</i>	.799	<i>.760</i>	<i>.716</i>	.932	.789	<i>.691</i>	.919	.773

Input description	Standard corporate description	Standard leisure description
O Tivoli Carvoeiro situa-se a 60 Km a Oeste do Aeroporto de Faro, na aldeia da Praia do Carvoeiro. Possui 293 quartos, Ar condicionado individual, TV satélite, telefone directo ao exterior, mini-bar, secador de cabelo, cofre e ADSL. O hotel dispõe de um café, uma piscina olimpica, Health Club com Sauna. Tem um parque Infantil. Também tem uma sala de reuniões equipada com ADSL.	Disponibiliza-se a cada hóspede cofre para a salvaguarda de pertences próprios. Cada quarto encontra-se equipado com ligação ADSL de alta velocidade e de uso gratuito. Nas instalações do nosso Hotel pode usufruir de uma sala de reuniões com toda a privacidade. Todos os quartos do nosso Hotel possuem ar condicionado para o seu conforto. A nível de localizações o Tivoli Carvoeiro situa-se a 60 Km a Oeste do Aeroporto de Faro, na aldeia da Praia do Carvoeiro.	A nível de localizações o Tivoli Carvoeiro situa-se a 60 Km a Oeste do Aeroporto de Faro, na aldeia da Praia do Carvoeiro. Todos os quartos encontram-se equipados com TV Satellite para seu entretenimento. Nas instalações do nosso Hotel pode usufruir parque infantil onde as suas crianças poderão encontrar toda a diversão. O Hotel possui ainda uma piscina Olimpica para os seus dias de Verão e para os dias de Inverno a sala de sauna encontra-se disponível à espera de uma visita sua.
		Standard English leisure description
		All hotel rooms have air conditioning. You may also enjoy our restaurant with a diverse set of menus and leave your children on the playground area for their amusement. The Hotel also has an Olympic pool for sunny days and a sauna room for cold winter days.

Fig. 3. An hotel description and three flavor descriptions generated by the system

5.3 System Evaluation

With the *Sentence Classifier* defined, and taking into account its future use, several tests were carried out using hotel descriptions residing on the Keyfor-Travel application. Fig. 3 shows an example of hotel description (in Portuguese) and the result of running the system with three different flavors of the *Ontology Translator*: a corporate and leisure templates for Portuguese and a leisure one for English.

For this example, the system was able to identify the hotel location and all the available equipment (8 ontology instances) and services (6 ontology instances)

and generate descriptions focusing on different sets of attributes according to the corporate or leisure flavor. For instance, while corporate description centers attention on internet access, meeting room and air conditioning, the leisure description spots the satellite TV, olympic pool and sauna.

6 Conclusions and Future Work

This paper presents a methodology for extracting useful information from textual descriptions of tourism products and standardizing it. This method was applied to hotel descriptions domain.

Results show that the main objective has been reached since it is possible to extract useful information, standardize it and create computable objects from plain text descriptions.

Concerning information extraction, this approach is able to detect relevant types of hotel descriptions, namely hotel services, equipment and location and, for each one, extract the useful features that describe them.

The possibility of having that information structured enables the creation of new added-value services, such as offer refinement search. Further, it provides the automatic creation of suitable descriptions for different market segments anticipating new steps towards a more effective client differentiation.

Regarding future work, and since there is room for improvements on system's various modules, we hope to increase its overall performance.

On the other way, its our aim to construct an hotel database, automatically populated and maintained by the application. This shall be used as a first "out-of-the-box" hotel repository, thus reducing substantially the effort on setting up an hotel infrastructure for commercial business.

Finally, we intend to address the areas of multi-language support (currently only Portuguese descriptions are supported) and normalization of other tourism products, such as rent-a-car and holiday packages.

References

1. Aitken, J.S.: Learning information extraction rules: An inductive logic programming approach. In: van Harmelen, F. (ed.) ECAI 2002 15th European Conference on Artificial Intelligence, Lyon, France, pp. 355–359 (2002)
2. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 100–110 (1999)
3. Development, H.: Jena – A Semantic Web Framework (March 2010), <http://jena.sourceforge.net>
4. Freitag, D., McCallum, A.: Information extraction with hmm structures learned by stochastic optimization. In: AI 2000 17th National Conference on Artificial Intelligence, pp. 584–589. AAAI Press (2000)
5. Grau, J.: Travel Agencies Online. eMarketer (2005)
6. Grishman, R.: Information Extraction: Techniques and Challenges. In: Pazienza, M.T. (ed.) SCIE 1997. LNCS, vol. 1299, pp. 10–27. Springer, Heidelberg (1997)

7. Hobbs, J.R., Bear, J., Israel, D., Tyson, M.: Fastus: A finite-state processor for information extraction from real-world text. In: IJCAI 1993 13th International Joint Conference on Artificial Intelligence, pp. 1172–1178 (1993)
8. Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML 1999 16th International Conference on Machine Learning (1999)
9. Klein, D., Manning, C.D.: Conditional structure versus conditional estimation in nlp models. In: ACL 2002 Conference on Empirical Methods in Natural Language Processing, pp. 9–16 (2002)
10. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML 2001 18th International Conference on Machine Learning, pp. 282–289 (2001)
11. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10, 707–710 (1966); originally publish in Russian
12. Martin, A., Przybocki, M. (eds.): 2003 NIST Language Recognition Evaluation (2003)
13. McCallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In: AAI 1998 Workshop on Learning for Text Categorization (1998)
14. Mladenić, D., Grobelnik, M.: Feature selection for unbalanced class distribution and naïve Bayes. In: ICML 1999 16th International Conference on Machine Learning, pp. 258–267 (1999)
15. Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Schölkopf, B., Burges, C., Smola, A. (eds.) *Advances in Kernel Methods – Support Vector Learning*, pp. 185–208. MIT Press (1999)
16. Quinlan, R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo (1993)
17. Salton, G., Wang, A., Yang, C.: A vector space model for information retrieval. *Journal of the American Society for Information Retrieval* 18, 613–620 (1975)
18. Schütze, H., Hull, D., Pedersen, J.: A comparison of classifiers and document representations for the routing problem. In: *SIGIR 1995 18th ACM International Conference on Research and Development in Information Retrieval*, Seattle, US, pp. 229–237 (1995)
19. Shawe-Taylor, J., Cristianini, N.: *Kernel Methods for Pattern Analysis*. Cambridge University Press (2004)
20. Tong, R., Appelbaum, L.: Machine learning for knowledge-based document routing. In: Harman (ed.) *TREC 2002 2nd Text Retrieval Conference* (1994)
21. Vapnik, V.: *Statistical learning theory*. Wiley, NY (1998)
22. ViaTecla: KEYforTravel platform (March 2010), <http://www.keyfortravel.com>
23. Voorhees, E. (ed.): *MUC7, 7th Message Understanding Conference*. Science Applications International Corporation (SAIC), Fairfax, Virginia (1998)
24. W3C: *OWL Web Ontology Language Guide* (March 2010), <http://www.w3.org/TR/owl-guide>
25. Witten, I., Frank, E.: *Data Mining: Practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Improving Collaborative Filtering in Social Tagging Systems

Felice Ferrara and Carlo Tasso

University of Udine, Via delle Scienze 206, 33100 Udine, Italy
{felice.ferrara,carlo.tasso}@uniud.it

Abstract. User-based Collaborative Filtering (CF) systems generate recommendations for a specific user by combining feedback (i.e. information about what is relevant for a user) provided by a set of people similar to that user. In these system the similarity among people is computed by taking into account the set of shared resources. However, there are several application domains, such as social tagging systems, where each user may have several different Topic of Interests (ToIs). In these cases, two users could share only some interests and, therefore, only a part of the feedback should be considered for producing recommendations. Focusing on social tagging systems, we propose here a novel approach to detect ToIs in the collection of the bookmarks of a user. Given a specific ToI, we adaptively identify similar people (i.e., sharing the same ToI) and select only the resources relevant to the specific ToI.

Keywords: Social tagging, collaborative filtering, adaptive system, personalization.

1 Introduction

Recommender systems are technologies aimed at facing the information overload problem: the massive amount of information available on the Web makes harder and harder for a user the task of finding relevant contents. In particular, Collaborative Filtering (CF) recommender systems take into account the opinions of people in order to filter relevant information. More specifically, given a user, a CF recommender system uses the feedback provided by people with similar interests in order to identify what is relevant for the specific user. In other words, a CF recommender simulates the behavior of humans: when people need to take a decision without a complete knowledge of all possible choices they are used to ask suggestions to others.

In order to automatize this task, a user-based CF recommender system exploits two main steps which we will refer in this paper as *neighbor selection* and *resource filtering*. More specifically, during the neighbor selection phase the system identifies the set of people which share interests with the target user (often referred as *active user*). In order to reach this aim a CF system has to compare the behavior of the users, and more technically, their user profiles. According to the specific application domain, several possible strategies can be adopted to

generate the user profiles, however the rationale behind the neighbor selection phase never changes: people which provide a similar feedback share (with an higher probability) the same interests. Several metrics have been used to compare the feedback of users and by using such techniques the neighbor selection phase can filter the set of *neighbors*, i.e. the set of people which share interests with the active user. Then, during the resource filtering phase, the feedback of neighbors is combined in order to identify the set of new potentially relevant resources to be suggested to the active user. This phase is based on the idea that people which had similar opinions in the past could consider as relevant the same resources also in the future. For this reason the resources evaluated positively by a larger set of neighbors are considered more relevant than others.

Such technologies have been applied also in social Web applications, like social tagging systems, in order to simplify the access to available resources. Social tagging systems are particularly interesting due to the fact that they allow the users to annotate relevant resources by means of personal *tags*. The set of annotated resources of a user (*personomy*) can be browsed by the specific user as well as by other users. This means that the set of all personomies, referred as *folksonomy*, is a pool of distinct personal perspectives which can be browsed by other users in the community in order to satisfy personal (potentially different) information needs. The semantic relations added by social annotation process is the main feature which allows the users to browse the folksonomy and, in this paper, we propose to use these social semantic relations to overcome some limitations of both the neighbor selection phase and the resource filtering phase described above.

More specifically, the neighbor selection step compares the feedback of users without taking into account that each user may have several interests and consequently the feedback also can be related to various different interests. This lowers the precision of this phase since a user could share an interest with a set of neighbors, but it could share other interests with completely different neighborhoods. To face this issue, we propose in this paper to take into account the social semantic relation built by the active user in his personomy for identifying his Topic of Interests (ToIs). Moreover, we assume that the active user can exploit various sets of tags in order to describe resources related to various information needs. By focusing on the set of resources associated to specific set of tags we find users interested in the specific ToI: in this way we identify a specific neighborhood according to the specific ToI. In other words, we propose a CF system which simulates an aspect of the human behavior which is not modeled by the CF schema previously described: humans are used to ask suggestions to a specific group of people accordingly to the specific information need. Moreover, since the neighbors also may have various interests we also filter the feedback of the neighbors according to the specific ToI considered.

The paper is organized as follow: in Section 2 related work is discussed, the construction of the user profile is shown in Section 3, the mechanism used to compute recommendations is described in Section 4, in Section 5 we compare

our approach with a baseline CF recommender system, Section 6 focuses on a variation on the approach, conclusions and future works are illustrated in Section 7.

2 Related Work

Web 2.0 tools promote the collaborative work of users: people can share, create, and classify resources by using Web applications. Social tagging tools, in particular, allow the users to annotate resources (identified by an URI) according to their personal criteria by means of tags. Large samples of annotated resources have been analyzed in several works with the aim of extracting social/semantic relations among tags, i.e. to infer semantic similarities among tags by evaluating the socially generated classifications. In order to reach this aim two main approaches have been proposed: (i) approaches which take into account co-occurrences among tags [1]; (ii) approaches which associate tags to concepts defined in other knowledge source such as WordNet or Wikipedia [2].

Several works (which we previously surveyed in [3]) integrate these social semantic relation in recommender systems. More specifically, several CF recommender systems analyze the ternary relation involving users, tags, and resources in order to extract social semantic relations among tags by evaluating co-occurrences among tags. Co-occurrences among tags can be observed at two levels of analysis: the folksonomy level and the personomy level.

By extracting similarities from the folksonomy, a system can infer the similarity between tags according to the co-occurrences among tags on the entire set of available resources. We exploited this approach in a framework [4] which models the user interests by grouping ‘similar’ tags utilized by the active user where the similarity among two tags depends on the number of times the two tags co-occur on the same resources in the entire folksonomy. This measure of similarity among tags is then used to compute the relevance. However, this approach infers relations among tags by taking into account the annotations provided by all users in the community discarding information about how the active user specifically combines tags. In fact, a folksonomy merges the annotations provided by the entire community without taking into account specific personal interests and tagging strategies of different users.

On the other hand, a personomy embodies information strictly related to the personal interests of a user. This means that the analysis of a personomy can reveal relations among tags which may be meaningful only for that user. The analysis of the personomy of the active user has been exploited in [5] where the authors propose a CF recommender system which catches semantic relations provided (implicitly through tagging) by the active user. More specifically, a community detection algorithm is used to group tags that the active user applied frequently together: each user is modeled by sets of tags and the similarity among users is computed by evaluating the similarity among the sets of tags they used. The main drawback of this approach is that users which share an interest could use different sets of tags to describe the same concept.

The approach described in this paper analyzes the personomy of a user in order to discover the set of his interests. However, differently from the approach described in [5], we recognize that users can apply the same tags to describe distinct concepts as well as they can use different tags to refer to the same concept. For this reason, we also filter the feedback of neighbors by taking into account only resources which are more strictly related to the specific ToI of the active user.

The idea of identifying neighborhood in an adaptive way was proposed in an item-based recommender system described in [6]: in order to assess the relevance of an item this approach filters the feedback of the active user by taking into account only the items which are both relevant for the active user and more similar to the specific item.

3 Detecting the User Interests

Our approach is based on the assumption that the active user exploits different sets of tags for indexing resources associated to different interests. This does not mean that the identified interests are not in a relations, but it is reasonable to assume that the user exploits a specific set of tags in order to distinguish some resources from the others.

More specifically, we model the user interests as a set of ToIs $(ToI_{au}^1, \dots, ToI_{au}^k)$.

The k -th ToI is defined by a set of weighted tags $T_{au}^k = \{(t_1, w_{t_1}^k), \dots, (t_n, w_{t_n}^k)\}$ and a set of weighted resources $R_{au}^k = \{(r_1, w_{r_1}^k), \dots, (r_m, w_{r_m}^k)\}$. More specifically, T_{au}^k is defined on a set of semantically related tags $tag(T_{au}^k) = \{t_1, \dots, t_n\}$ previously utilized by the active user, where two tags are considered to be in a semantic relation if the active user has applied them together to classify one or more resources. The weight associated to each tag represents the relevance of the tag with respect to that ToI and it is used to compute the relevance of each resource $res(R_{au}^k) = \{r_1, \dots, r_m\}$ tagged by the active user within that ToI.

The way the active user combines tags is accounted in order to identify semantic relationships among tags. In particular, the personomy of the active user is modeled by an undirected weighted graph P where: each node represents a tag; an edge connects two tags if they have been used together to label one or more resources; an edge connecting two tags is weighted by the number of times two tags have been used together.

In Figure 1, we show the graph P created to model the personomy of a user in the BibSonomy dataset (<http://www.kde.cs.uni-kassel.de/bibsonomy/dumps>), where both we take into account the bookmarks added during the last 6 months and we do not show the weights associated to edges in order to make the graph readable.

The graph representation of a personomy is the input of a graph clustering algorithm aimed at grouping tags with a shared semantic. In particular, we follow the idea proposed in [7] where a node (representing a tag in our model) may be in more than one cluster identifying, in this way, overlapping clusters of tags. This clustering technique identifies clusters of tags by identifying subgraphs from

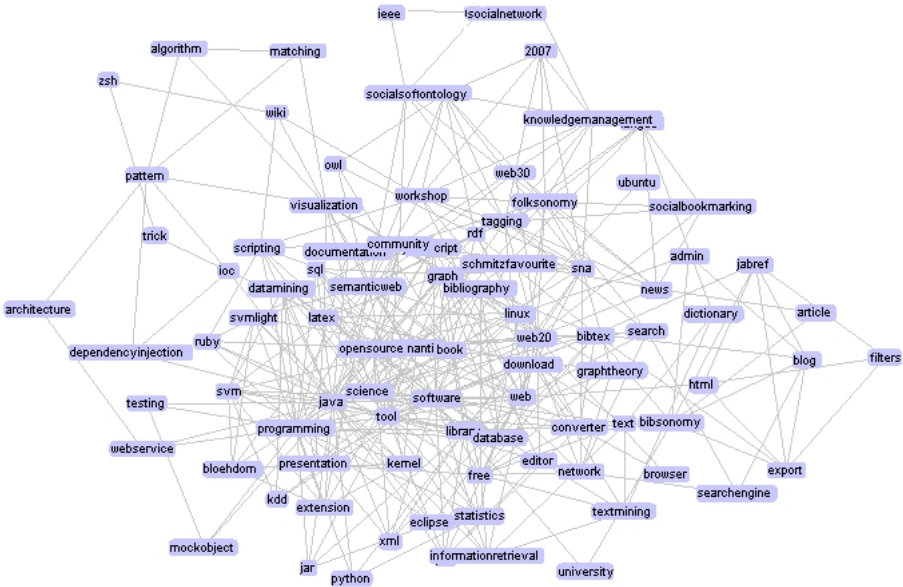


Fig. 1. An example of the graph P for a user of the BibSonomy system

the starting graph P , where each subgraph G maximizes the following *fitness property* f_G :

$$f_G = \frac{K_{in}}{(K_{in} + K_{out})^\alpha}$$

where K_{in} is the sum of the weights of the edges which connect two tags belonging to G , K_{out} is the sum of the weights which connect tags in G with the rest of the graph, and α is a parameter which controls the size of clusters. By using this metric, the fitness of a subgraph increases when we add to the subgraph a tag that the user has exploited frequently in co-occurrence with tags contained in the subgraph and rarely with the others. The algorithm builds, for a given starting node, a cluster of tags by adding, at each step, the node which maximizes the following fitness function $f_G^a = f_{G+a} - f_{G-a}$, where $G + a$ is the subgraph obtained by adding the node a to the subgraph G and $G - a$ is the subgraph obtained by removing the node a to the subgraph G . The process which adds tags to the cluster stops when there are not tags with a positive fitness value. Lower values for the α parameter generate wider clusters, higher values of the α parameter generate clusters of tags more closely related to the starting node.

Using this method, our approach starts by building the cluster associated to the most used tag. Then, the approach identifies the cluster for the most used tag which has not been yet included in a cluster, and so on. The clustering algorithm terminates when each tag is at least in one cluster. At the end, each subgraph detected by the clustering phase contains the set of tags associated to a certain ToI for the active user.

However, given a subgraph G^k , some of the tags in G^k are less relevant than others since they possibly are used also for referring to other different ToIs. Therefore, we associate a weight w_t^k to each tag t in the subgraph G^k by computing the betweenness centrality [8] of the node associated to the tag t in the specific subgraph identifying, in this way T_{au}^k : given a subgraph, the betweenness centrality measures the centrality of the nodes (the tags) in the specific subgraph.

Applying this strategy on the example shown above for the tag ‘java’ with $\alpha = 1.0$ the ToI will contain the following set of tags (ordered according to the weight assigned to each of them): *java, documentation, mockobject, jar, schmitz-favourite, xml, ruby, scripting, testing, eclipse, programming, python, webservice, opensource, statistics, library, javascript, tool, sql, database, software, graph, community, kdd, informationretrieval, linux, datamining, university, download, converter, bibtex, bibliography, text, textmining, dictionary, free*. On the other hand, starting the computation of the cluster from the tag ‘folksonomy’ we obtain the following cluster of tags: *folksonomy, tagging, socialbookmarking, socialsoftware, ontology, search, semantic, knowledgemanagement, workshop, 2007, news, langde, searchengine, article, web20, socialnetwork, web30, community, semanticweb, web, sna, ieee, owl, rdf, network, wiki, graphtheory*.

The set T_{au}^k is used to infer the set R_{au}^k such that $res(R_{au}^k)$ is the set composed by the resources that the active user labeled by tags in $tag(T_{au}^k)$ and the weight w_r^k for the resource r is equal to the maximum weight of the tags belonging to the set $tag(T_{au}^k)$ which the active user assigned to r .

4 Recommending Resources for a ToI

This section focuses on the recommendation process by describing how the approach filters and ranks resources for a specific $ToI_{au}^k = (T_{au}^k, R_{au}^k)$ of the active user. We will show how the set of weighted resources R_{au}^k can be used to select adaptively the neighbors (Section 4.1) and then how feedback from neighbors is filtered and combined (Section 4.2).

4.1 Adaptive Neighbor Selection

Given the $ToI_{au}^k = (T_{au}^k, R_{au}^k)$ of the active user, the set of weighted resources R_{au}^k is used to filter the set of neighbors for the ToI. In particular, the approach identifies people interested in the specific ToI by taking into account only the users who tagged the resources in $res(R_{au}^k)$. We assume that people interested in ToI_{au}^k share with the active user relevant resources within the specific ToI. For this reason, let $R_{shared}(u, R_{au}^k)$ be the set of resources that the user u share with the active user in $res(R_{au}^k)$, we compute how much the specific interest of the active user is matched by the neighbor u by computing the following *InterestMatch*

$$InterestMatch(u, ToI_{au}^k) = \frac{\sum_{r_i \in R_{shared}(u, R_{au}^k)} w_{r_i}^k}{\sum_{r_i \in res(R_{au}^k)} w_{r_i}^k}$$

The rationale of the *InterestMatch* is that higher is the number and the relevance of the resources in R_{au}^k that the neighbor u have also tagged, and higher is the interest of u in the specific ToI. By using the *InterestMatch* function, we can select the set N_{au}^k of the top N neighbors interested in ToI_{au}^k .

4.2 Filtering and Combining Feedback for the ToI

The neighbor selection phase takes in account only resources in the ToI of the active user. In order to produce recommendations we need to identify new resources labeled by neighbors which are related to the specific ToI. Therefore, to achieve this goal, we consider the tags that the neighbor u applied on the set of shared resources $R_{shared}(u, R_{au}^k)$: the resources labeled by u with these tags are considered relevant for the specific ToI. We follow the idea that, some tags in the personomy of the neighbor u are more trustworthy than others for finding relevant resources for the ToI. In fact also the neighbor may have several ToIs and, for this reason, we are interested in discovering which tags utilized by u better match for the ToI of the active user. We consider more trustworthy the tags which have been used by the neighbor to label many relevant resources within ToI_{au}^k and, specifically, we compute a measure of the *trustworthiness* of a tag t_j in the collection of the neighbor u with respect to ToI_{au}^k as follow:

$$trustworthiness_u(t_j, ToI_{au}^k) = \frac{\sum_{r_i \in R_{shared}(u, R_{au}^k)} w_{r_i}^k \cdot \phi(u, t_j, r_i)}{\sum_{r_i \in R_{shared}(u, R_{au}^k)} w_{r_i}^k}$$

where $\phi(u, t_j, r_i) = 1$ if the user u has applied the tag t_j on the resource r_i , 0 otherwise. Following the principle that trustworthy tags are associated to relevant resources of the neighbor u , we assign an higher relevance to resources labeled by more trustworthy tags. Specifically, we compute $rel_u(r_j, ToI_{au}^k)$, which is the relevance of the resource r_j in the personomy of the neighbor u with respect to ToI_{au}^k , as the highest trustworthiness associated to tags that the neighbor u assigned to r_j .

Finally, the relevance of a resource r_j for the active user with respect to ToI_{au}^k is computed by summing the relevance of r_j over the collections of the neighbors N_{au}^k as follow:

$$rel(r_j, ToI_{au}^k) = \sum_{u \in N_{au}^k} InterestMatch(u, ToI_{au}^k) \cdot rel_u(r_j, ToI_{au}^k)$$

This allows to produce the ranked list of resources that are recommended to the active user.

5 Evaluation

In this work, we are interested in evaluating if the proposed approach improve the accuracy of traditional CF systems, i.e. if the idea of producing a distinct

set of recommendations for each ToI of the user can improve the accuracy of a CF system. To this aim, we developed a baseline CF recommender system where tagging information is discarded: the user profile is constituted by a unary vector over the set of the resources in the system (the user profile has a 1 for a resource if she tagged the resource, 0 otherwise); the similarity among two users is measured by computing the cosine similarity among the two unary vectors associated to the users; the relevance of a resource to the active user depends on the similarity of the N neighbors most similar to the active user. We exploited an off-line evaluation by using the BibSonomy dataset where we basically compared the results provided by the baseline approach to the results computed by our approach when only the main ToI of the active user (the ToI most accessed) is considered. More specifically, we divided the BibSonomy dataset into two chunks: the training set which includes all bookmarks until the first of January 2008; the test set which has the bookmarks from the first of January 2008 to the 31 December 2008. We created the user profile only for the users who tagged at least 80 resources until the first of January 2008 and who tagged at least 10 resources during the 2008. This is reasonable since CF approaches produce effective recommendations only when users rated a significant number of items [9].

The quality of the computed recommendations produced by the two mechanisms have been evaluated by adopting the *hit-rate* (HR) measure. The HR measure, which has been described and used also in [10] to compare two CF recommender systems, is defined as follow

$$hit-rate = \frac{Number\ of\ hits}{m}$$

where m is the total number of users considered in the evaluation and we count a hit when the system produces at least one correct recommendation (i.e. a recommendation for a resource that the active user has actually tagged in the subsequent period). Given the lists of recommendations for the m users produced by a recommendation mechanism, the hit-rate is a value in $[0, 1]$ which is higher when there is a larger number of users who received at least one recommendation for a resource that they will tag in the test period. Both the baseline approach and the tag based approach produced 10 recommendations using feedback from the top 10, top 20, and top 30 neighbors and the results are shown in Table 1.

Table 1. Hit-rate with 10, 20 and 30 Neighbors

	HR (10 Nei)	HR (20 Nei)	HR (30 Nei)
Baseline	0.20%	0.23%	0.23%
Tag Based (main ToI)	0.26%	0.41%	0.44%

The table shows that by generating recommendations for the main ToI only the proposed approach can better satisfy the informative needs of users. We are currently working on techniques for integrating the entire set of ToI into the

computation of recommendation. This task is very complex since distinct ToIs may be semantically related: the user may use distinct set of tags for describing similar concepts.

6 On the Generation of the ToIs

In this work we compare our approach with a baseline CF recommender system in order to prove that the proposed approach is reasonable better. In particular, in order to evaluate the results of our approach we generated the recommendations only for the cluster generated by starting from the tag that the active user more frequently utilized. However, during the first experimentations we observed that the choice of using the most used tag for generating the clusters is not always the best one. In fact the most used tag sometimes has not a clear semantic (the reader can think for instance to some generic tags such as *all* or *toread*). Obviously, this can lower the precision of the approach since the identification of a ToI strongly depends on the specific starting tag used to generate the cluster.

In order to face this limitation we are working on alternative techniques aimed at extending the described approach by detecting the most meaningful clusters in the personomy. At the moment, we are evaluating an approach which generates a ToI starting from each tag in the personomy and then it analyzes the generated ToIs in order to find the most meaningful one.

More specifically, we generate a ToI starting from each tag in the personomy. These ToIs are then used to compute the similarity among the tags in the personomy. In particular, we compute the distance among two tags t_a and t_b as

$$sim(t_a, t_b) = \frac{|ToI(t_a) \cap ToI(t_b)|}{|ToI(t_a) \cup ToI(t_b)|}$$

where $ToI(t_x)$ is the set of the ToIs which contain the tag t_x . This distance takes into account the number of times that two tags appear in the same clusters: more frequently two tags appear in the same clusters higher is the similarity between the specific tags.

By using this distance we can modify the clustering procedure: instead of starting from a single tag we use the most similar pair of tags to generate a ToI. Then, the next pair of tags used to generate a cluster is the most similar pair of tags where at least one of the tag has not been included in the previous clusters, and so on.

The rationale of this alternative approach is that tags which are used in different ToI are also present in different clusters. On the other hand, it is plausible that tags which more frequently appear in the same clusters are more strictly related. We have implemented a prototype of this extension and, at the moment, we are carrying on some experiments.

7 Conclusions

In this work we describe a CF recommender system which takes into account the personomy of a user to adaptively construct his user profile. We compare the

described approach with a baseline CF recommender system in order to verify the plausibility of the approach. Preliminary results show that the proposed approach is reasonable and, at the moment, we are planning a more effective evaluation to validate this claim by comparing our approach to other competitive systems such as the system described in [1]. We also propose a possible extension of the described work aimed at producing more accurate user profiles.

Future work will also focus on: (i) extending the approach in order to identify hierarchical organization of user interests; (ii) adding a more semantic layer by means of content/ontology based analysis.

References

1. Zanardi, V., Capra, L.: Social ranking: Finding relevant content in web 2.0. In: Proc. of the 2nd ACM Int. Conf. on Recommender Systems, Lausanne, Switzerland (2008)
2. Cantador, I., Szomszor, M., Alani, H., Fernández, M., Castells, P.: Enriching Ontological User Profiles with Tagging History for Multi-Domain Recommendations. In: Proceedings of the 1st International Workshop on Collective Semantics: Collective Intelligence and the Semantic Web (CISWeb 2008), pp. 5–19 (2008)
3. Dattolo, A., Ferrara, F., Tasso, C.: On social semantic relations for recommending tags and resources using folksonomies. In: Human-Computer Systems Interaction. Backgrounds and Applications 2
4. Dattolo, A., Ferrara, F., Tasso, C.: Supporting Personalized user Concept Spaces and Recommendations for a Publication Sharing System. In: Houben, G.-J., McCalla, G., Pianesi, F., Zancanaro, M. (eds.) UMAP 2009. LNCS, vol. 5535, pp. 325–330. Springer, Heidelberg (2009)
5. Zhou, T., Ma, H., Lyu, M., King, I.: Userrec: A user recommendation framework in social tagging systems. In: Proc. of the 24th AAAI Conf., Atlanta, Georgia, USA, pp. 1486–1491 (2010)
6. Baltrunas, L., Ricci, F.: Locally Adaptive Neighborhood Selection for Collaborative Filtering Recommendations. In: Nejdl, W., Kay, J., Pu, P., Herder, E. (eds.) AH 2008. LNCS, vol. 5149, pp. 22–31. Springer, Heidelberg (2008)
7. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* 11 (2009)
8. Opsahl, T., Agneessens, F., Skvoretz, J.: Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks* 32 (2010)
9. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *Adaptive Web 2007*. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007)
10. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. *Transaction on Information Systems* 1, 143–177 (2004)

Ontology-Driven Method for Integrating Biomedical Repositories

José Antonio Miñarro-Giménez and Jesualdo Tomás Fernández-Breis

Faculty of Computer Science, University of Murcia, Spain
{jose.minyarro, jfernand}@um.es

Abstract. Handling the increasing number of biological repositories and the growing amount of information they contain is a significant challenge for researchers. Thus, Semantic Web Technologies are the basis of the new approaches that are dealing with such challenges. In this paper, we describe a methodology to manage the integration of biomedical repositories into a knowledge base. This method is based on the explicit definition of the mappings between relational resources and the domain ontology and the conditions that determine the identity of a particular instance. We will describe both the method and its application to the OGO system which integrates orthologous genes and genetic diseases repositories into an ontological knowledge base.

Keywords: Ontology-driven Integration, Ontological Knowledge Base, Biological repositories, Orthologous Genes, Genetic Diseases.

1 Introduction

The current situation of the public database infrastructure provides a very large collection of heterogeneous biological databases which is constantly increasing [1]. This makes the research on new methodologies, which can handle the new problems for their integration and computational processing, necessary. Initially, small research communities defined their own data structures, organization and vocabularies. The major limitation of such approach was that the biological data could not be efficiently used and shared with other communities. Thus, bioinformatic repositories, such as NCBI Entrez [2] or UniProt [3], were designed to compile such disperse information and to provide a common reference to them.

Due to the terminological heterogeneity, some initiatives have been launched to develop common vocabularies among different work groups. For example, the Gene Ontology (GO) [4], which is one of those initiatives, reduces the semantic heterogeneity associated to the annotation of gene products between different databases. An ontology is a formal, explicit specification of a shared conceptualisation [5], which provides a shared vocabulary and can be used as a domain model. The success of GO provoked a huge interest in designing, developing and using biomedical ontologies, whose number has rapidly increased [6]. Projects such as the OBO Foundry [7] promote the development and use of biomedical ontologies.

From a technical perspective, ontologies are the cornerstone technology for the Semantic Web [8], which is an extension of the current World Wide Web, in which the semantics of information and services on the web are well defined. This represents an evolution in which the web content is understandable by both humans and machines. In fact, different Semantic Web technologies such as RDF [9], OWL [10] or SPARQL [11] have been used for developing semantic biomedical solutions.

The majority of public biomedical databases provide its content for download by using flat files, so the heterogeneity of information, the loose definition of file structure and the wide variability of data are the main causes which make their integration difficult, and therefore, hinders the development of unattended applications. This situation hampers the definition of mapping rules because the formal description of its elements is not available. The meaning of the fields is often described in text files. Relational databases are also used to store biomedical data, but this solution also lacks a formal definition of their tables and columns meaning. However, relational databases provide the repository schema which facilitates its use for defining the mapping rules and performing the integration among them.

Our purpose is to develop a method for the automatic execution of biomedical data integration processes based on the formalization of the relations between the different biomedical resources that are integrated. This method would then support the development of integrated biomedical semantic systems, since the result of the integration process is an integrated, semantic repository. In this paper, we will describe both the method and the initial results we have obtained in the application of the integration method to our OGO system [12]. OGO is a semantic repository of orthologs [13] and genetic diseases built by integrating various bioinformatic repositories by using a hard-coded integration method. This lack of formalization causes maintenance problems in the semantic repository that we hope to overcome with this enhanced method.

The structure of this paper is described next. First, a brief description of the OGO system is presented in Section 2. Then, Section 3 describes the method designed for integrating the information source. In Section 4, we describe the application of the method to OGO. Finally, some conclusions are put forward in Section 5.

2 The OGO System

The OGO system integrates biological repositories about orthologous genes and genetic diseases in order to allow biomedical researchers to access to a more complete, integrated and non-redundant information. OGO integrates the information of orthologs available in the following resources: KOG [14], Homologene [15], OrthoMCL [16] and Inparanoid [17]. The information about genetic diseases is collected from OMIM [18].

The OGO system is based on the OGO ontology (see Figure 1), which conceptualizes the orthologs and genetic disease domains. This ontology reuses the

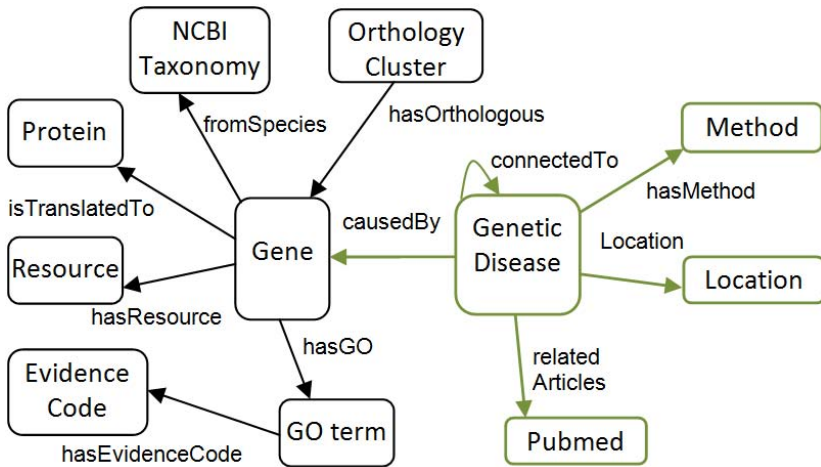


Fig. 1. The OGO ontology

following well-known bio-ontologies: GO [4], Evidence Code Ontology [19], NCBI taxonomy [20] and Relationship Ontology [21].

The amount of information and the wide variety of information sources make the management of the integration process a complex task. Consequently, having an unsupervised approach that enables us to automatically update the OGO knowledge base becomes of paramount importance. It should be noted that the different source repositories are updated at different times, so the integration process should be not applied to all the resources at the same time. In addition to this, changes in the repositories or in the ontology should have minimum impact in the integration process. These are the main problems that we identified in the maintenance method used to date in the OGO system and that we aim to solve with the method presented in this paper.

3 The Integration Method

This methodology is designed to integrate relational repositories into an ontological knowledge base. Thus, this method permits the association of database schemas with ontology models. Figure 2 describes graphically the connections between the different parts involved in the integration process. First, the module *Mapping Rules Definition* is used to define the *Mapping Rules* between the *Relational Repositories* and the *Ontology Model*. Second, the module *Identity Rules Definition* is used to define the policy for detecting duplicate instances. If there exists an equivalent instance, then both instances have to be merged to avoid redundancy. Therefore, we need the *Identity Rules* module to describe the instances of ontology classes. Finally, the *Integration Module* uses the mapping rules and the information of the repositories to create the instances in the

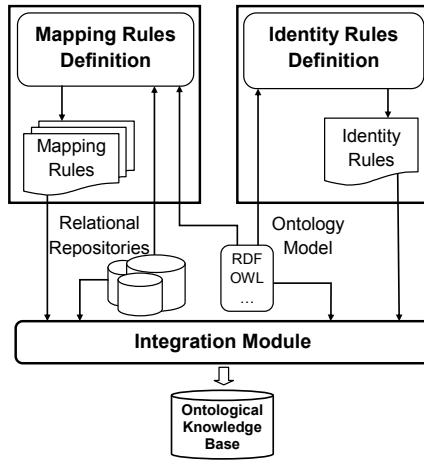


Fig. 2. Methodology schema

ontology model, and the identity rules to decide whether the instances should be added into the *Ontological Knowledge Base*.

3.1 Mapping Rules

The mappings rules define the connections between the repositories schemas and the ontology resources. The rules are coded by using XML documents and each document contains the mapping rules between a particular schema and ontology. In each XML document, the repository is described by using the "`<dbsource>`" tag and the ontology is described by using the "`<ontotarget>`" tag. These tags define the parameters which will be used to create a connection to the repository and load the ontology during the integration task. In particular, the repository connection parameters, that are defined in "`<dbsource>`" tag are (1) the *type* of repository; (2) *connection*, which indicates the URL and the particular schema where the data is located; (3) the name of the *user account* in the repository; and (4) the *password* of the user account. The current implementation supports MySQL, PostgreSQL and Oracle.

There are three types of mapping rules. The first type represents the mapping between a key of the repository and an ontology class, and it is identified by the "`<type>DB2Class</type>`" value. The second type represents the mapping between a table and a property of the ontology, and its identifier is "`<type>DB2Prop</type>`". The third type is the mapping between a table and a relationship of the ontology, and it is identified by the "`<type>DB2Rel</type>`" value.

Table 1 shows an example of the first type of mapping rules. This rule consists of the class of the ontology and the related key column of the repository. The class is defined in the "`<class>`" tag by using its URI in the ontology, and the key

column is defined in the ”<db>” tag by using the ”table_name.column_name” nomenclature. This rule relates a key column to a class in the ontology, so each key value will be used to identify a specific instance of the ontology class and to gather the other properties of the instance from the repository. If a class of the ontology has already defined all its instances, this kind of rule will not be needed. The instances would be identified in a proper way in the other types of rules by using the same key or other foreign key.

Table 1. DB2Class type of mapping rule for the Gene class

```
<map>
<type>DB2Class</type>
<class><id>http://miuras.inf.um.es/ontologies/OGO.owl#Gene</id></class>
<db><id>geneinf.geneId</id></db>
</map>
```

Table 2 describes the second type of mapping rules, which relates a property of the ontology with the columns of the repository in which the information is stored. This rule consists of *source*, *predicate* and *target* tags. The *source* tag describes the key to identify the appropriate instances that are related to the property. It consists of ”<class>” and ”<db>” tags which have the same meaning as in the *DB2Class* type. The *predicate* tag identifies a datatype property or an annotation property of the ontology by using its URI. Finally, the *target* tag defines where to find the property values. It uses ”table.column” nomenclature as described above.

Table 2. DB2Prop type of mapping rule for the *Gene_identifier* property

```
<map>
<type>DB2Prop</type>
<source>
<class><id>http://miuras.inf.um.es/ontologies/OGO.owl#Gene</id></class>
<db><id>geneinf.geneId</id></db>
</source>
<predicate>
<id>http://miuras.inf.um.es/ontologies/OGO.owl#Gene_identifier</id>
</predicate>
<target>
<db><id>geneinf.geneId</id></db>
</target>
</map>
```

The last type of mapping rule is described in Table 3. This rule provides a description of how two instances of the ontology are linked by using information from the repository. In this example, we relate gene instances and Gene Ontology instances through *participates_in*. To describe how to link the information the rule uses three tags *source*, *predicate* and *target*, such as *DB2Prop* does. The *source* tag identifies the subject instance of a relationship using the ”<class>” and ”<db>” tags. The *predicate* tag identifies the relationship.

So, the ”<property>” tag identifies the root relationship in the ontology and ”<db>” tag identifies the specific relationship, which is a sub-property of the root one. In order to associate the ontology subproperties with the values of a repository column, the subproperty and the column must share a tag. The *target* tag identifies the object instance of the relationship. As well as the *source* tag, the *target* tag uses the ”<class>” and ”<db>” tags to identify instances in the ontology.

This type of rules cover all type of information from the repositories that we need to load into the knowledge base. However, these rules do not take into account any criteria for the integration of information and hence we need identity rules to eliminate redundancy in the knowledge base.

Table 3. DB2Rel type of mapping rule for the *participates_in* relationship

```

<map>
<type>DB2Rel</type>
<source>
<class><id>http://miuras.inf.um.es/ontologies/OGO.owl#Gene</id></class>
<db><id>gene2go.genelId</id></db>
</source>
<predicate>
<property><id>http://miuras.inf.um.es/ontologies/OGO.owl#participates_in</id></property>
<db><id>gene2go.ec</id></db>
</predicate>
<target>
<class><id>http://um.es/go.owl#GO_0003674</id></class>
<db><id>gene2go.goId</id></db>
</target>
</map>

```

3.2 Identity Rules

The aim of identity rules is to find which instances refer to the same real world individual. Such rules describe the requirements that two instances have to meet to be considered equivalent. Hence, before adding a new instance into the Knowledge Base, we look for any instance with similar meaning using the requirements defined in these rules. To match an instance in the Knowledge Base, the requirements have to be assessment considering its boolean operators as well. Identity rules are coded by using XML documents. Each identity rule is related to a class in the ontology. So, we have one identity rule document for each ontology. The identity rules file also contains the reference to the ontology file. The ”<ontosource>” tag is used for defining its location. Each identity rule in the file is represented within the ”<condition>” tag.

In particular, an identity rule is determined by a tree of requirements. The root indicates which class of the ontology is described in the rule. Requirements are described within the ”<requirement>” tag and contain the description of a (subject, predicate, object) relationship. There is one ”<predicate>” tag and another ”<object>” tag included within each requirement tag, but the subject’ of the relationship is defined by the object value of the parent node in the tree of requirements. So, requirements are connected through parent-child and sibling relationships.

The parent-child relationships between requirements are classified into "AND" and "OR" tags, whose meaning are, respectively, mandatory and optional. As usual, the precedence of "AND" is higher. The predicate consists of "<scope>" and "<property>" tags. The values of the scope tag are "ALL" and "SOME". They determine whether all property values of an instance must exist or only some of them. The property tag indicates the property to be evaluated.

On the other hand, the object tag consists of "<value>" and "<class>" tags. The values of the value tag are *EQUALS* and *EQUALS IGNORE CASE*. It defines how to evaluate the property values. If the datatype of the property is a string both type of values are possible. In other cases, only the *EQUALS* value is possible. The "<class>" tag specifies the class of the range of the property when using Object Properties. In other cases, it identifies the class of the domain of the property.

Table 4 shows the identity rules defined for the Gene class. The URIs of the resources of the ontology have been simplified for clarity, we only show their local names. Thus, the root requirement contains the object tag, which indicates the ontology class to be depicted. Then, there are two "and" tags, the first one means that the each sub-requirement is mandatory, but the second refers to the requirements between two siblings. So, the first sub-requirement indicates that two equivalent genes must belong to the same organism, whereas the second sub-requirement is empty, without "predicate" or "object" tags. These requirements are used to group other requirements and to define the precedence between them. Finally, an empty requirement contains two sub-requirements that are optional because it contains the "or" tag. These requirements indicate at least a name or an identifier of the gene, respectively, that should match.

Table 4. Example of the identity rule for the Gene class

```

<condition>
  <requirement><object><class>Gene</class></object>
  <and><and>
    <requirement>
      <predicate><scope>ALL</scope><property>fromSpecies</property></predicate>
      <object><value>EQUALS</value><class>Organism</class></object>
    </requirement>
    <requirement>
      <or>
        <requirement>
          <predicate><scope>SOME</scope><property>Gene_name</property></predicate>
          <object><value>EQUALS IGNORE CASE</value><class>Gene</class></object>
        </requirement>
        <requirement>
          <predicate><scope>SOME</scope><property>Gene_identifier</property></predicate>
          <object><value>EQUALS</value><class>Gene</class></object>
        </requirement>
      </or>
    </requirement>
  </and></and>
</requirement>
</condition>

```

As the integrated biomedical data is stored in a semantic repository, SPARQL is the most appropriate query language. The evaluation of the identity rule is

Table 5. SPARQL query related to an identity rule

```

PREFIX ogo: <http://miuras.inf.um.es/ontologies/OGO.owl#>
PREFIX tax: <http://um.es/ncbi.owl#>
SELECT ?subject
WHERE {
  ?subject ogo:fromSpecies tax:organism1 .
  {
    { ?subject ogo:Gene_name ?label1 FILTER regex(?label1, "name1", i) . }
    UNION { ?subject ogo:Gene_name ?label2 FILTER regex(?label2, "name2", i) . }
    UNION { ?subject ogo:Gene_identifier "id" . }
  }
}

```

done by translating its requirements into SPARQL. Table 5 shows an example of SPARQL defined using the identity rule shown in Table 4. This query searches for an URI of an instance which belongs to "*organism1*" and contains the gene names "*name1*" or "*name2*", or the gene identifier "*id*".

3.3 Integration Process

The integration process uses the rules for mapping the information from the repositories to the knowledge base. Each repository has its own mapping rules document, but the document of identity rules is associated with the global ontology of the knowledge base.

To integrate a repository, first we have to parse the identity rules file and its corresponding mapping rules file. Then, we have to group the mapping rules with its corresponding the class. Afterwards, we have to sort out those groups to avoid dependences through relationships. Each group contains the rules for gathering the information of the instances of its class. In particular, the "DB2Class" rule indicates the key values to retrieve all instances. If one group of mapping rules does not contain that type of rule, it means that the instances of such class do not have to be created because they already exist in the knowledge base. Once retrieved the information of an instance, we first check whether there exists an equivalent instance in the knowledge base. If not, we add it.

4 Results

We have put into practice the methodology defined by means of integrating the orthologous genes and protein information into the OGO KB. In this evaluation, we have not integrated all the repositories. The process has focused on two of the resources, namely, KOG and Homologene. KOG provides *circa* 5000 clusters of orthologous genes whilst Homologene contains *circa* 20000 clusters. Both repositories were integrated into the OGO KB, resulting in nearly 95000 genes instances.

The mapping and identity rules were defined using a desktop application. This application provides a graphical interface to define the mapping and identity rules. As a result of this application we have a mapping rules file for KOG repository and another for Homologene. We also have defined a identity rule file related to the OGO model.

After performing the integration, we compared the results with the ones obtained by following the previous integration method used by us in OGO. Apart from benefits of the automation of the process, the results were even better since the method was able to detect some equivalent individuals that the previous method was not. The identity rules provide a more precise definition of the instances and their properties to detect equivalent instances in comparison to the manual integration which only take into account the instance identifiers. Thus, these results are encouraging and we plan to apply the method to the other source repositories.

5 Conclusions

The definition of a methodology for mapping relational repositories into an ontological repository is needed to facilitate the integration of repositories. However, the lack of formalization of such integration process would make the integrated data hard to maintain and update. In this paper we have proposed a method based on explicit mapping and identity rules and in which the meaning of the entities is defined in an ontology.

On the one hand, mapping rules are focused on how to relate the source information with the ontology. Mapping rules depend on repository schemas and ontology models, so any change on them might result in a redefinition of the rules.

On the other hand, identity rules are focused on defining what makes instances different each other. This situation provides enough flexibility to handle both heterogeneous information sources and complex ontological models. Identity rules depend only on the semantics of the domain, so they are not affected by changes in the source repositories. The quality of a knowledge base is improved by the identity rules definition. Thus, an instance is not only identified through an identifier but also through its properties and relationships. The identity rules also avoid redundancy and inconsistency, because they contain some of the restrictions defined in the ontology model. The consistency of the KB is evaluated through the use of reasoners over the integrated repository. However, it should be noted that an inconsistency in the KB might be caused by a badly defined identity rule.

The method has been successfully applied to a part of the OGO system and we are going to apply it to the whole OGO. We will also do further research in the relation of identity rules with the notion of keys in OWL [22] and the notion of identity in Ontoclean [23].

Acknowledgments. This project has been possible thanks to the funding of the Spanish Ministry of Science and Innovation through grant TIN2010-21388-C02-02.

References

1. Galperin, M., Cochrane, G.: The 2011 nucleic acids research database issue and the online molecular biology database collection. *Nucleic Acids Research Advance Access Nucl. Acids Res.* 39, D1–D6 (2011)

2. NCBI: Ncbi entrez global search portal, <http://www.ncbi.nlm.nih.gov/Entrez/>
3. Uniprot Consortium: Ongoing and future developments at the universal protein resource. *BMC Bioinformatics* 39, 214–219 (2010)
4. Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., Harris, M., Hill, D., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J., Richardson, J., Ringwald, M., Rubin, G., Sherlock, G.: Gene ontology: tool for the unification of biology. *Nature Genetics* 25, 25–29 (2000)
5. Gruber, T.: Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies* 43, 256–274 (1995)
6. Bodenreider, O., Stevens, R.: Bio-ontologies: current trends and future directions. *Briefings in Bioinformatics* 7, 256–274 (2006)
7. Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L., Eilbeck, K., Ireland, A., Mungall, C., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S., Scheuermann, R., Shah, N., Whetzel, P., Lewis, S.: The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotech.* 25(1087-0156), 1251–1255 (2007)
8. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* 284(5), 34–43 (2001)
9. RDF Working Group: Resource description framework, <http://www.w3.org/RDF>
10. W3C: Owl web ontology language, <http://www.w3.org/TR/owl-features>
11. W3C: Sparql query language for rdf, <http://www.w3.org/TR/rdf-sparql-query>
12. Miñarro Giménez, J.A., Madrid, M., Fernández-Breis, J.T.: Ogo: an ontological approach for integrating knowledge about orthology. *BMC Bioinformatics* 10(S-10), 13 (2009)
13. Fitch, W.M.: Distinguishing homologous from analogous proteins. *Systematic Zoology* 19(2), 99–113 (1970)
14. Tatusov, R., Fedorova, N., Jackson, J., Jacobs, A., Kiryutin, B., Koonin, E., Krylov, D., Mazumder, R., Mekhedov, S., Nikolskaya, A., Rao, B., Smirnov, S., Sverdlov, A., Vasudevan, S., Wolf, Y., Yin, J., Natale, D.: The cog database: an updated version includes eukaryotes. *BMC Bioinformatics* 4, 41+ (2003)
15. NCBI: Homologene database, <http://www.ncbi.nlm.nih.gov/homologene>
16. Chen, F., Mackey, A.J., Stoeckert Jr., C.J., Roos, D.S.: Orthomcl-db: querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acid Research* 34, 363–368 (2005)
17. Remm, M., Storm, C., Sonnhammer, E.: Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of Molecular Biology* 314(5), 1041–1052 (2001)
18. NCBI: Omim database, <http://www.ncbi.nlm.nih.gov/omim/>
19. The Gene Ontology: Evidence codes ontology, <http://www.obofoundry.org/cgi-bin/detail.cgi?id=evidencecode>
20. NCBI: Ncbi taxonomy ontology, http://obofoundry.org/cgi-bin/detail.cgi?ncbi_taxonomy
21. Smith, B., Ceusters, W., Klagges, B., Kohler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A., Rosse, C.: Relations in biomedical ontologies. *Genome Biology* 6, R46 (2005)
22. W3C: Easy Keys, http://www.w3.org/2007/OWL/wiki/Easy_Keys
23. Guarino, N., Welty, C.: Evaluating ontological decisions with ontoclean. *Communications of the ACM* 45(2), 61–65 (2002)

Lightweighting the Web of Data through Compact RDF/HDT

Javier D. Fernández¹, Miguel A. Martínez-Prieto^{1,2}, Mario Arias¹,
Claudio Gutierrez², Sandra Álvarez-García³, and Nieves R. Brisaboa³

¹ Universidad de Valladolid, España
{jfergar,migumar2}@infor.uva.es, mario.arias@gmail.com

² Universidad de Chile, Chile
cgutierrez@dcc.uchile.cl

³ Universidade da Coruña, España
{salvarezg,brisaboa}@udc.es

Abstract. The Web of Data is producing large RDF datasets from diverse fields. The increasing size of the data being published threatens to make these datasets hardly to exchange, index and consume. This scalability problem greatly diminishes the potential of interconnected RDF graphs. The HDT format addresses these problems through a compact RDF representation, that partitions and efficiently represents three components: Header (metadata), Dictionary (strings occurring in the dataset), and Triples (graph structure). This paper revisits the format and exploits the latest findings in triples indexing for querying, exchanging and visualizing RDF information at large scale.

1 Introduction

The Web of Data comprises very large RDF datasets from diverse fields such as bioinformatics, geography or social networks. The Linked Data Project has been playing a crucial role promoting the use of RDF and HTTP to publish structured data on the Web and to connect it between different data sources. This philosophy has lift traditional hyperlinks to a new stage, in which more than 25 billion RDF triples are being shared and increasingly linked. Linked Open Data (LOD) cloud is roughly doubling every 10 months, hence the important problem when these data need to be managed.

To date, these RDF datasets tend to be published, exchanged and consumed within plain RDF formats such as RDF/XML, N3 or Turtle, which provide human-focused syntaxes disregarding large data volumes. General compressors are used over these plain formats in order to reduce the final size, but the resultant files must be decompressed and parsed in plain at the final consumer.

¹ <http://www.w3.org/TR/REC-rdf-syntax/>

² <http://linkeddata.org>

³ <http://www4.wiwiwiss.fu-berlin.de/lodcloud/>

⁴ <http://www.w3.org/DesignIssues/Notation3>

⁵ <http://www.w3.org/TeamSubmission/turtle/>

Several RDF indexes and RDF storages explore efficient SPARQL⁶ query resolution methods [154]. However, these approaches suffer from lack of scalability [21]. There is still a large interest in querying optimization [19], whose performance is diminished when the RDF storages manage these very large datasets.

All this is diminishing the potential of interconnected RDF graphs due to the huge space they take in and the large time required for consuming. Thus, only a small portion of the data tend to be finally exchanged, indexed and consumed.

RDF/HDT (Header-Dictionary-Triples) addresses these issues. It proposes a binary format for publishing and exchanging RDF data at large scale [10]. This paper revisits RDF/HDT and analyzes its role in the Web of Data. We provide a set of application fields which need to overcome the aforementioned scalability problems, studying RDF/HDT in such contexts. In particular, we focus in querying, exchanging and consuming RDF, *i.e.*, the consuming usage of large RDF information. To this later end, we refer to a novel tool which consumes HDT to provide large RDF data visualization.

The paper is organized as follows. Section 2 reviews the underlying problems of large RDF in the Web of Data. Section 3 presents an overview of HDT concepts and practical issues of their implementation. We revisit HDT for indexing and querying in Section 4, studying two different solutions for Triples indexing. We provide an HDT-based architecture for RDF exchanging in Section 5. Section 6 encourages HDT for RDF consumption at large scale, referring to a visualization tool as a use-case. Finally, Section 7 concludes and addresses future challenges.

2 Related Work

The RDF data model was designed as a general framework for the description and modeling of information, hence it is not attached to a particular serialization format. RDF/XML, despite its verbosity, is useful for interchanging small-scale data. Other notations, *e.g.* Turtle and N3, allow shortening some constructions, such groups of URIs or common datatypes. However, none of these proposals seems to have considered data volume as a primary goal.

Although diverse techniques provide RDF indexes, the efficient and scalable resolution of SPARQL remains an open problem. Some of them store RDF in a relational database and perform SPARQL queries through SQL, *e.g.* Virtuoso⁷. A specific technique, called vertical-partitioning, groups triples by predicate and defines a 2-column (S,O) table for each one [21]. They allow some SPARQL queries to be speeded up, but make some others difficult, *e.g.* the queries with unbounded predicates. A different strategy is followed in RDF-3X [15] and BitMat [4]; indexes are created for all ordering combinations (SPO, SOP, PSO, POS, OPS, OSP), increasing spatial requirements.

The access points of the Web of Data, built on top of RDF, are typically the SPARQL Endpoints, services which interpret the SPARQL query language. The performance of querying this infrastructure is diminished by the aforementioned factors [18]: (1) the **response time**, affected by the efficiency of the RDF indexing structure, and (2) the overall **data exchange time**, obviously influenced by the serialization format.

⁶ <http://www.w3.org/TR/rdf-sparql-query/>

⁷ <http://www.openlinksw.com/dataspace/dav/wiki/Main/VOSRDF>

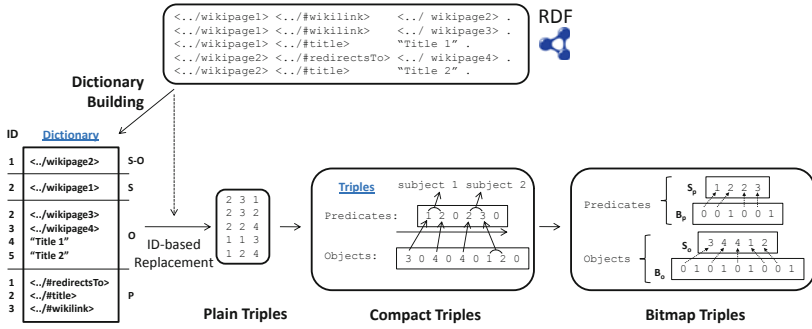


Fig. 1. Incremental representation of an RDF dataset with HDT

SPARQL resolution over the Web of Data has been addressed in two different ways. On the one hand, a federated query architecture, in the sense of traditional databases federation [20], sets up an abstraction layer of multiple SPARQL Endpoints [17]. Decomposition of queries, subquery propagation and integration of results are its main challenges. On the other hand, data centralization is based on dataset replication under a unique access point, e.g. the well-known Sindice service [16] or the Linked Data aggregation of OpenLink Software[8]. Both mechanisms suffer from a problem of dynamic data discovery [12] and large data management and indexing.

3 RDF/HDT

Traditional formats for serializing RDF stay influenced by the *old* document-centric perspective of the Web. This leads to fuzzy publications, inefficient management, complex processing and lack of scalability for large RDF datasets. The format RDF/HDT (*Header-Dictionary-Triples*) arises as a compact alternative to the plain formats for serializing RDF in the current Web of Data, moving forward to a data-centric scheme.

3.1 Basic Concepts

RDF/HDT [10] is a binary format for RDF recently published as a W3C Member Submission [9]. It considers the skewed structure of large RDF datasets [22] to achieve large spatial savings. It splits a dataset into three logical components:

- **Header.** This component is an RDF graph expressing metadata about the dataset. It extends Void [19] with a specific vocabulary [11] which allows logical and physical descriptions for the dataset. It can be used through well-known mechanisms, such as SPARQL Endpoints, serving as an entrance point to the information described in the dataset.

⁸ <http://lod.openlinksw.com/>

⁹ <http://www.w3.org/Submission/2011/SUBM-HDT-20110330/>

¹⁰ <http://www.w3.org/2001/sw/interest/void/>

¹¹ <http://www.rdfhdt.org/hdt/>

Table 1. Compression results and Triples sizes for several datasets

Dataset	Triples (millions)	Size (GB)	Compression (MB)				Triples Size (MB)	
			gzip	bzip2	ppmd	HDT-C	Bitmap	k ² -Triples
geonames	9.4	1.00	78.54	54.78	49.15	32.36	33.60	17.41
wikipedia	47.0	6.88	491.04	360.01	288.85	156.40	143.84	124.93
dbtune	58.9	9.34	924.85	630.28	441.86	175.02	245.78	152.58
uniprot	72.5	9.11	1233.25	739.76	637.15	330.23	278.59	81.92
dbpedia-en	232.5	33.12	3513.58	2645.36	2251.95	1319.29	995.73	884.74

- **Dictionary.** It maps all different strings to integer IDs. This decision pursues the goal of *compactness* because each triple can be now regarded as a group of three integer IDs.

- **Triples.** This component represents the graph topology by encoding all triples in the dataset. The mapping of the Dictionary allows the structure to be managed as an integer-stream. This new representation facilitates the encoder to take advantage of the existing power-law distributions for subjects and objects [10], improving HDT effectiveness.

3.2 Practical Issues

RDF/HDT supports a flexible implementation for each component depending on the final application consuming RDF. Figure 1 provides an example of different practical strategies. First, the Dictionary is built from the original RDF dataset. It is implemented on a simple hashing-based approach which distinguishes strings playing roles of: shared subject-object (S-O), subject (S), object (O) and predicate (P). Then, these mappings are used to describe three different techniques for the Triples [10].

Plain triples is the most naive approach in which only the ID replacement is carried out. **Compact triples** performs a subject ordering and creates predicate and object adjacency lists for each subject. The stream *Predicates* concatenates the predicate lists related to each subject, using the non-assigned zero ID as separator. The second stream (called *Objects*) lists all objects related to the pairs (s, p) in the same way. Finally, **Bitmap triples** extracts the auxiliary zero IDs embedded in each stream and stores them in two bitmaps in which 1-bits mark the end of the corresponding adjacency list.

Fernández, *et al.* [10] also proposes HDT-Compress, which combines specific compression techniques for the Dictionary and the Triples. Table 1 studies the effectiveness of this approach and compares them against well-known compressors for several datasets [2]. As can be seen, HDT-Compress (column HDT-C) always achieves the best compression ratio. The comparison of HDT-Compress against the typical compressor used in the area (gzip) reports improvements from 2.5 up to more than 5 times. The difference against the best compressor (ppmd) is reduced, from 1.5 to 2.5 times, but remains significant. These results support HDT as a very compact serialization format for RDF and encourage its usage in applications, such as exchanging or publishing, in which the dataset size determines their efficiency.

¹² Geonames, dbtune and uniprot (<http://km.aifb.kit.edu/projects/btc-2010>), wikipedia (<http://labs.systemone.at>) and dbpedia (<http://wiki.dbpedia.org>)

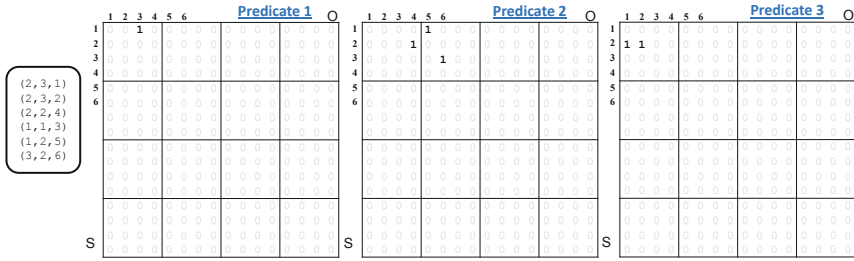


Fig. 2. k^2 -triples: vertical partitioning on k^2 -trees

4 RDF Indexing and SPARQL Querying

RDF indexing is a cornerstone of the Web of Data because it determines the performance of SPARQL resolution, and so, the efficiency of other tasks such as *reasoning*. A common weakness for current RDF indexing solutions is the significant time that they waste in disk transfers. Although full-in-memory indexes seem a logical solution, they are hardly scalable due to their lack of compression. In this scenario, HDT arises as an effective solution because of its compactness. This section addresses two HDT-based approaches focused on the indexing of the Triples component.

4.1 Bitmap Triples

This is an intuitive technique based on the Bitmap triples representation described in the previous section. Let us suppose a dataset containing $|S|$, $|P|$, and $|O|$ different subjects, predicates and objects respectively. Each predicate, in S_p , is represented with $\log(|P|)$ bits whereas each object, in S_o , takes $\log(|O|)$ bits. In turn, the bitmaps B_p and B_o are also represented in plain form [14]. This technique uses a 5% of extra space over the original bitmap length in order to achieve efficient constant time for rank and select operations [14]. These two operations enable graph structure traversing and allow some SPARQL triple pattern queries to be performed [13].

Regarding the *SPO* ordering, Bitmap triples resolves efficiently the triple patterns (S, P, O) , $(S, P, ?O)$, $(S, ?P, ?O)$, and $(S, ?P, O)$. Note that all of them bound the subject. Patterns with unbounded subject require additional indexes to be resolved.

4.2 k^2 -Triples

The *Bitmap triples* approach achieves an interesting tradeoff between compression and searching features, but it is not a full-index by itself. As we explained, the ordering chosen for its building restricts the queries than can be efficiently answered. However, its performance yields an important conclusion: compression allows RDF indexes to be fully managed in main memory, achieving very efficient SPARQL resolution.

¹³ Triple patterns are like RDF triples except that each of the subject, predicate and object may be a variable. ?X values are used to indicate variable elements in the pattern.

The usage of compact data structures [14] is not very common in semantic applications. However, they have been successfully used to solve problems in areas such as Bioinformatics [7] or Web Graphs [6]. A technique from this last domain, called k^2 -tree, has been generalized to be used for representing general graph databases [1]. It models a graph as a binary matrix in which a cell (i, j) contains 1 iff the nodes i and j are linked. This technique supports very effective resolution for queries which (1) retrieves all points in a row x (direct neighbours for x); (2) retrieves all points in a column y (reverse neighbours for y); (3) checks the existence of a given point (x, y) ; and (4) performs a bidimensional range queries involving subsets of rows and/or columns.

k^2 -triples [2] uses k^2 -trees to compress and index the Triples component of an RDF/HDT dataset. It is, to the best of our knowledge, the first RDF index built on compact data structures. It vertically partitions the dataset to group all triples related to a given predicate. This decision allows each group to be modeled with an independent k^2 -tree which indexes all pairs (subject, object) associated with a given predicate. The resulting k^2 -trees describe very sparse 1 distributions which allow k^2 -triples to achieve ultra-compressed representations of the Triples component.

Figure 2 shows how k^2 -triples represents the listed triples, extended from the example in Figure 1. Three independent k^2 -trees are used for indexing the triples. Note that this approach works on square matrices, hence the rows/columns are expanded. In the example, only the three first rows (for the three existing subjects) and the six first columns (for the six objects) are really used in each k^2 -tree, so all triples are stored in these ranges. For instance, the predicate 2 takes part in three triples: $(2,2,4)$, $(1,2,5)$ and $(3,2,6)$, and its corresponding k^2 -tree stores them in the coordinates $(2,4)$, $(1,5)$ and $(3,6)$, which represent the corresponding subject-object pairs.

k^2 -triples supports all SPARQL triple pattern queries on the primitive operations of the k^2 -tree. The conjunction of these patterns allows more complex queries to be obtained through join conditions. It currently gives support for subject-subject joins, object-object and cross-joins between subjects and objects.

The results reported for k^2 -triples [2] give three interesting facts: 1) it achieves the most compressed representations¹⁴; 2) it largely outperforms vertical partitioning on relational databases; and 3) it beats multi-index solutions for queries with bounded predicates. These results support k^2 -triples for the design of full-in-memory RDF engines and its performance excels for datasets using a limited number of predicates.

5 RDF Exchanging

Communication processes in the Web of Data are threatened by the overall data exchange time. Even if current RDF formats are compressed using universal techniques, they must be decompressed at destination and then parsing the same verbose data.

HDT combines compressibility (as stated in Section 3.2) and cleaner parsing, since it already provides an index to the information. Besides establishing a compact RDF binary format for exchanging, HDT can be used as basis to design an efficient architectural model in the Web of Data. The state-of-the art reveals the need of improving the efficiency of SPARQL Endpoints, supporting (1) efficient and scalable mechanisms for

¹⁴ Table 1 also shows that k^2 -triples outperforms Bitmap triples compressibility.

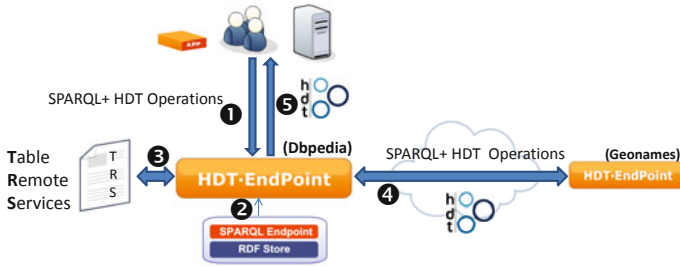


Fig. 3. Structure and communication flow in HDT-Endpoints

storing and indexing large RDF datasets, (2) compact formats for exchanging, such as HDT, and (3) protocols for discovering new resources.

HDT-Endpoints is an architecture which extends the concept of SPARQL Endpoints to support HDT functionality, taking advantage of its properties to overcome the mentioned needs. The net is built on top of HDT-Endpoint nodes.

Definition 1 (HDT-Endpoint). *An HDT-Endpoint node is an element (i) conforming to the SPARQL protocol for RDF (SPROT)¹⁵, (ii) which extends its functionality to discover and communicate with other HDT-Endpoints, and (iii) makes use of HDT as its RDF interchange format.*

Figure 3 shows the structure and communication flow for two HDT-Endpoints, storing Dbpedia and Geonames in the LOD cloud. Imagine a client, (e.g. a human, a machine or a consuming application), who wants to retrieve all the information about “Berlin”. She will send a SPARQL query to the Dbpedia HDT-Endpoint (step ‘1’ in the figure) which tries to solve it locally (‘2’). Then, the Dbpedia node will look up in its Table of Remote Services (‘3’) to discover other HDT-Endpoints which could contribute in the results. It will discover Geonames, send a subquery (‘4’), harvest the results (sent in HDT) and present the final result (also in HDT) to the user (‘5’).

The Table of Remote Services is a mechanism to discover other HDT-Endpoints through the HDT Header. This can be seen as a “routing” table, which includes one entry per HDT dataset held in the HDT-Endpoint. Each entry stores its namespaces and the URI of the HDT-Endpoint hosting the dataset. It also includes the Header of the dataset and an optional timestamp in order to support an updating policy.

In addition to SPARQL Queries, HDT-Endpoints allows for specific HDT operations, such as returning all (of a part of) each components (Header, Dictionary, Triples).

6 Consuming RDF. Large RDF Visualization

At this point, consuming RDF is seriously underexploited [13] due to (1) the huge RDF graphs exchanging costs, (2) their complex parsing and indexing and (3) a general darkness of the underlying structure. Large RDF data tend to be complex and hard to

¹⁵ <http://www.w3.org/TR/rdf-sparql-protocol/>

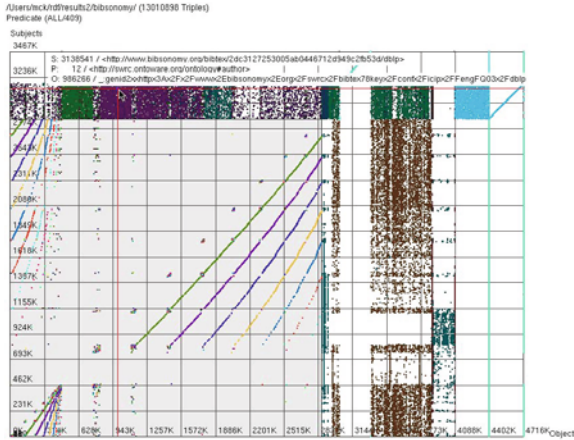


Fig. 4. Bibsonomy dataset as shown using the HDT visualization tool

read/parse in its textual publication format. Thus, semantic information developers have to deal with painful processes in order to consume these large RDF graphs.

RDF/HDT leads to compact RDF representations which not only mitigate exchanging costs, but also make the parsing and indexing easier. This way, applications consuming HDT can benefit from the reduced size as well as “instant” access to the data.

A visualization tool is proposed [3] as an example of application consuming HDT. In addition, it provides a solution for the third aforementioned problem for consumers, *i.e.* visualization and understanding of large RDF data.

6.1 Background

The use of visual tools helps consumers understand RDF content. Some of the typical tasks are identifying the most relevant resources in the graph, whether the information is grouped or scattered, or browsing the links between resources. Typical visualization tools use the node-link representation of the underlying graph. Since huge RDF datasets contain thousands to millions of statements, the number of graph nodes and edges [8] is large, causing users to have trouble interpreting the visualization. A completely different approach for rendering graph data is using its adjacency matrix [9]. It consists of generating a boolean-valued connectivity table where rows and columns represent the vertices of the graph, and each cell (x,y) states whether x is connected to y or not.

6.2 Adjacency Matrix Visualization

Arias et al. [3] proposes using a 3D adjacency matrix as an alternate visualization method for RDF. The RDF data must be available in HDT beforehand, so that the compact information can be directly consumed by the application.

The Triples component of HDT, which represents each statement as a three integer triple, can be seen as a (x, y, z) coordinate in a 3D space that can be plotted as point

in a 3D scattered plot. The y axis represents subjects, the x axis objects and the z axis predicates. The user can rotate and zoom the view to have different 3D perspectives of the data. The first and most interesting view is the one that places the camera on the z axis looking at the origin, showing a 2D figure comparing subjects against objects (Figure 4). Predicates are also highlighted using a different color for each one.

Each axis scale is annotated using the IDs, so the user can get a first sight of the amount of subjects and objects. The shared subject-object area of the dictionary is represented using a rectangle at the origin with a different background color. This area is quite interesting, because it represents the links among RDF resources.

The user experience can be enhanced by providing some extra features for interactively browsing the data. The user can hover the mouse above the graphic, showing details of the nearest triple under the cursor. The HDT Triples component can be queried to find the nearest triple, and finally the full triple can be converted back to string using the HDT Dictionary.

7 Conclusions and Future Work

RDF/HDT is a binary serialization format for RDF which decomposes the original data into three logical components: Header, Dictionary and Triples. It exploits the skewed structure of RDF datasets to achieve large spatial savings. Besides establishing a compact RDF format, HDT also provides efficient querying and parsing. We revisit HDT and study its applications in typical scenarios within the current Web of Data. We focus on indexing, querying, exchanging and consumption of large RDF datasets.

We analyze indexing and querying of HDT information through two different approaches for the Triples component, Bitmap and k^2 -triples. Bitmap triples is a compact representation suitable for scalable exchange, but it only supports basic query operations. K^2 -triples emerges as an ultra-compressed full-in-memory solution supporting complex SPARQL operations. Experiments show that k^2 -triples is the most effective technique among all considered solutions, and the most efficient engine for solving triple patterns with bounded predicates. For future work, a query optimizer integrated with HDT would allow more complex queries to be efficiently resolved. New dictionary implementations can be also explored for providing native searches over the data, allowing to compute SPARQL filters before the triples search.

The HDT·EndPoints architecture leads to mitigate the scalability problem of the current Web of Data by means of HDT exploitation; larger volumes can be managed with smaller delays, encouraging the distribution in the Web. Furthermore, the exchange of HDT, compact and searchable, allows for direct access to the information. For future, the analyzed features of RDF/HDT open a world of possibilities and applications in the Web of Data. In particular, we envision the use of this infrastructure in mobile devices. HDT would serve, not only as the RDF transmission format, but also as an internal storage and native indexing, due to its reduced size fits mobile devices constrains.

Regarding RDF consumption, the major strength of HDT is to deal with huge datasets, achieving efficient parsing and processing, as it already embeds an index to the information. We show its applicability to a concrete problem of visualizing large-scale RDF data. The tool, based on a 3D RDF adjacency matrix, consumes and makes

use of HDT to alleviate the limitations of previous node-link graph visualization approaches. RDF consumers can benefit from the latest findings in RDF/HDT. The logical decomposition of the original RDF in three components allows for different researches, implementation and improvements for future work.

Acknowledgments. This work is funded by the MICINN of Spain TIN2009-14009-C02-02 (first three authors), Junta de Castilla y León and the European Social Fund (first author) and Institute for Cell Dynamics and Biotechnology (ICDB), Grant ICM P05-001-F, Mideplan, Chile (second author); Fondecyt 1090565 and 1110287 (fourth author); and MICINN (PGE and FEDER) TIN2009-14560-C03-02, TIN2010-21246-C02-01 and CDTI CEN-20091048, Xunta de Galicia (cofunded with FEDER) ref. 2010/17 (fifth and sixth authors), and MICINN BES-2010-039022 (FPI program), for the fifth author.

References

1. Álvarez, S., Brisaboa, N., Ladra, S., Pedreira, O.: A Compact Representation of Graph Databases. In: Proc. of MLG, pp. 18–25 (2010)
2. Álvarez García, S., Brisaboa, N., Fernández, J.D., Martínez-Prieto, M.A.: Compressed k2-Triples for Full-In-Memory RDF Engines. In: Proc. of AMCIS, TBP (2011)
3. Arias, M., Fernández, J.D., Martínez-Prieto, M.A.: RDF Visualization using a Three-Dimensional Adjacency Matrix. In: Proc. of SemSearch (2011), <http://km.aifb.kit.edu/ws/semsearch11/8.pdf>
4. Atré, M., Chaoji, V., Zaki, M.J., Hendler, J.A.: Matrix “Bit” loaded: a scalable lightweight join query processor for RDF data. In: Proc of WWW, pp. 41–50 (2010)
5. Bizer, C., Heath, T., Idehen, K., Berners-Lee, T.: Linked Data On the Web (LDOW 2008). In: Proc. of WWW, pp. 1265–1266 (2008)
6. Brisaboa, N.R., Ladra, S., Navarro, G.: k^2 -Trees for Compact Web Graph Representation. In: Karlgren, J., Tarhio, J., Hyvärö, H. (eds.) SPIRE 2009. LNCS, vol. 5721, pp. 18–30. Springer, Heidelberg (2009)
7. Claude, F., Fariña, A., Martínez-Prieto, M.A., Navarro, G.: Compressed q -gram indexing for highly repetitive biological sequences. In: Proc. of BIBE, pp. 86–91 (2010)
8. Dokulil, J., Katreniakova, J.: RDF Visualization - Thinking Big. In: Proc. DEXA, pp. 459–463 (2009)
9. Fekete, J.: Visualizing networks using adjacency matrices: Progresses and challenges. In: Proc. of CAD/GRAPHICS 2009, pp. 636–638 (2009)
10. Fernández, J.D., Martínez-Prieto, M.A., Gutierrez, C.: Compact Representation of Large RDF Data Sets for Publishing and Exchange. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 193–208. Springer, Heidelberg (2010)
11. González, R., Grabowski, S., Makinen, V., Navarro, G.: Practical implementation of rank and select queries. In: Proc. of WEA, pp. 27–38 (2005)
12. Hartig, O., Bizer, C., Freytag, J.-C.: Executing SPARQL Queries over the Web of Linked Data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunaryan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 293–309. Springer, Heidelberg (2009)
13. Hogan, A., Harth, A., Passant, A., Decker, S., Polleres, A.: Weaving the Pedantic Web. In: Proc. of LDOW (2010)

14. Navarro, G., Mäkinen, V.: Compressed Full-Text Indexes. *ACM Computing Surveys* 39(1), article 2 (2007)
15. Neumann, T., Weikum, G.: The RDF-3X Engine for Scalable Management of RDF data. *The VLDB Journal* 19(1), 91–113 (2010)
16. Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., Tummarello, G.: Sindice.com: a document-oriented lookup index for open linked data. *International Journal of Metadata Semantics and Ontologies* 3(1), 37 (2008)
17. Quilitz, B., Leser, U.: Querying Distributed RDF Data Sources with SPARQL. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) *ESWC 2008*. LNCS, vol. 5021, pp. 524–538. Springer, Heidelberg (2008)
18. Schmidt, M., Hornung, T., Lausen, G., Pinkel, C.: SP²Bench: A SPARQL Performance Benchmark. In: *Proc. of ICDE*, pp. 222–233 (2009)
19. Schmidt, M., Meier, M., Lausen, G.: Foundations of SPARQL Query Optimization. In: *Proc. of ICDT*, pp. 4–33 (2010)
20. Sheth, A.P., Larson, J.A.: Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys* 22(3), 183–236 (1990)
21. Sidirourgos, L., Goncalves, R., Kersten, M., Nes, N., Manegold, S.: Column-store Support for RDF Data Management: not All Swans are White. *Proc. of the VLDB Endowment* 1(2), 1553–1563 (2008)
22. Theoharis, Y., Tzitzikas, Y., Kotzinos, D., Christophides, V.: On Graph Features of Semantic Web Schemas. *IEEE Trans. on Know. and Data Engineering* 20(5), 692–702 (2008)

Query Expansion Methods and Performance Evaluation for Reusing Linking Open Data of the European Public Procurement Notices*

Jose María Álvarez¹, José Emilio Labra¹, Ramón Calmeau², Ángel Marín³,
and José Luis Marín³

¹ WESO Research Group-Universidad de Oviedo

² EXIS TI

³ Gateway Strategic Consultancy Services

{josem.alvarez,jelabra}@weso.es, ramon.calmeau@exis-ti.com,

{anmar,josmar}@gateway-scs.es

<http://www.weso.es>,

<http://www.exis-ti.com>,

<http://gateway-scs.es/>

Abstract. The aim of this paper is to present some methods to expand user queries and a performance evaluation to retrieve public procurement notices in the e-Procurement sector using semantics and linking open data. Taking into account that public procurement notices contain information variables like type of contract, region, duration, total value, target enterprise, etc. different methods can be applied to expand user queries easing the access to the information and providing a more accurate information retrieval system. Nevertheless expanded user queries can involve an extra-time in the process of retrieving notices. That is why a performance evaluation is outlined to tune up the semantic methods and the generated queries providing a scalable and time-efficient system. On the other hand this system is based on the use of semantic web technologies so it is necessary to model the unstructured information included in public procurement notices (organizations, contracting authorities, contracts awarded, etc.), enrich that information with existing product classification systems and linked data vocabularies and publish the relevant data extracted out of the notices following the linking open data approach. In this new LOD realm these techniques are considered to provide added-value services like search, matchmaking geo-reasoning, or prediction, specially relevant to small and medium enterprises (SMEs).

1 Introduction

In the European e-Procurement context there is an increasing commitment to boost the use of electronic communications and transaction processing by

* This work is part of '10ders Information Services project' (<http://rd.10ders.net/>) partially funded by the Spanish Ministry of Industry, Commerce and Tourism with code TSI-020100-2010-919, led by 'Gateway Strategic Consultancy Services' and developed in cooperation with 'EXIS TI' and WESO Research Group.

government institutions and other public sector organizations. The European Commission outlines the following advantages in the wider use of e-Procurement¹: increased accessibility and transparency, benefits for individual procedures, benefits in terms of more efficient procurement administration and potential for integration of EU procurement markets. TED² ('Tenders Electronic Daily') is the on line version of the 'Supplement to the Official Journal of the European Union', dedicated to European public procurement (1500 new procurement notices every day³) but an unified information system pan-European dealing with: 1) dispersion of the information; 2) duplication of the same notice in more than one source; 3) different publishing formats; 4) problems regarding to a multi-lingual environment and 5) aggregation of low-value procurement opportunities, is missing.

Obviously one of the most interesting domains to apply the Linking Open Data (LOD) approach is public procurement information published by governmental contracting authorities. In that sense, the growing commitment to the reuse of public sector information (PSI) and initiatives like semantic web, LOD and the use of Knowledge Organization Systems (KOS) provide building blocks for an innovative unified pan-European information system for the benefit of SMEs.

This work aims to apply some semantic-based methods to expand user queries in the e-Procurement sector using semantic web technologies and the LOD approach. In this paper a survey of methods is presented to expand the information variables extracted out of the public procurement notices and a performance evaluation of the user queries is also provided to show how the system works. This study is motivated by the following example: *Which public procurement notices are relevant to Dutch companies (only SMEs) that want to tender for contracts announced by local authorities with a total value lower than 170K € to procure "Construction work for bridges and tunnels, shafts and subways" and a two year duration in the Dutch-speaking region of Flanders in Belgium?*

2 Related Work

In the scope of LOD and open government data (OGD) there are projects trying to exploit the information of public procurement notices like LOTED⁴ ("Linked Open Tenders Electronic Daily") where they use the RSS feeds of TED. UK government⁵ is doing a great effort to promote its information sources using the LOD approach. They have published datasets from different sectors: transport, defense, NUTS geographical information⁶, etc. Most of the public administrations in the different countries are also betting for LOD approach to make public

¹ http://ec.europa.eu/internal_market/consultations/docs/2010/e-procurement/green-paper_en.pdf

² <http://ted.europa.eu/>

³ <http://www.eubusiness.com/tenders>

⁴ <http://loted.eu:8081/LOTED1Rep/>

⁵ <http://data.gov.uk>

⁶ <http://nuts.psi.enakting.org/>

their information: Spain (Aporta project⁷), USA⁸, etc. Regarding the use of LOD and organizations there is a new ontology for modeling the information about organizations⁹ and recently it has been released “The Open Database Of The Corporate World”¹⁰.

Product Scheme Classifications (also known as PSCs) like the CPV (Common Procurement Vocabulary available at RAMON, the Eurostat’s metadata server) have been built to solve specific problems of interoperability and communication in e-commerce¹⁰. The aim of a PSC is to be used as a standard *de facto* by different agents for information interchange in marketplaces². Any PSC, as well as other classification systems can be interpreted as: 1) domain-ontologies⁹ or 2) conceptual schemes¹⁶ comprised of conceptual resources. Finally, Good Relations¹¹ is an ontology for the e-commerce developed by Martin Hepp et. al.

The use of semantic methods to exploit the data from the semantic web like Spreading Activation (SA) techniques and Rule Based Systems (RBSs) is widely used. The main application of SA techniques is focus on Document and Information Retrieval⁷. These techniques have been also used in semantic search based on hybrid approaches^{13,4}, user query expansion combining metadata and user information to improve web data annotations. RBSs have been used a long time to decision support, diagnosis, etc. in different fields. In the semantic web area and due to the apparition of OWL 2-RL, SPARQL Rules! and RIF these systems are growing in their use to deal with the web of data but a clear approach to mix datasets and RBSs is missing. They can also be applied to SA techniques to handle the activation and propagation of the concepts.

Finally the process of expanding queries is widely accepted to reformulate a seed query and improve retrieval performance in information retrieval operations. In most of the cases the process deals with linguistic issues³ through the use of controlled vocabularies and taxonomies to find synonyms, spelling errors, etc. In the case of e-Procurement a search engine should be able to process the user query and perform a concept based query expansion process like⁴ for legal documents.

3 Survey of Methods to Expand User Queries

The selection of methods to expand the information of a user query about public procurement notices depends on some factors: 1) the type of variable: concept from a taxonomy or ontology, a numeric value in a range or geographical information; 2) the intentions of the user by means of creating a search profile (RDF-based and reusing existing LOD and e-Procurement vocabularies) containing the initial selected values for the information variables presented in the notices

⁷ <http://www.aporta.es/>

⁸ <http://www.data.gov/>

⁹ <http://www.epimorphics.com/web/category/category/developers/organization-ontology>

¹⁰ <http://opencorporates.com/>

¹¹ <http://www.heppnetz.de/projects/goodrelations/>

Table 1. Survey of Methods to Expand User Queries in the e-Procurement sector

Variable	Type	User Intention	Statistical Information	Method	Tool
CPV and NUTS codes	Concept	Enhance Codes	Correlation among codes	<ul style="list-style-type: none"> • Syntactic comparison of descriptions • Dividing an initial value into narrower codes • Recommender • SA • Geo-reasoning 	<ul style="list-style-type: none"> • Apache Lucene • Hard-Coding • Apache Mahout • ONTOSPREAD (API Java for SA) • Geonames and GeoSPARQL
<ul style="list-style-type: none"> • Total Value • Duration and Publishing year • ... 	Numeric	Establish a Range	Correlation with historical information	<ul style="list-style-type: none"> • Numeric range • FuzzyLogic 	<ul style="list-style-type: none"> • Hard-Coding • JFuzzyLogic
Type of company	Enumerate		Correlation with historical information	Get type of companies for the CPV codes, etc.	Hard-Coding

and 3) the statistical information available in previous public procurement notices. Taking into account these factors Table 1 shows a comparison of the selected methods to be applied in the process of query expansion. The current situation of searching public procurement notices consists on the interaction between a business user and a client that wants to tender with a certain set of restrictions on the information variables. However the intentions of the client do not match with the real information in notices that is why the business user must rewrite client restrictions to convert them in a real query that can retrieve the desired notices. These expansion methods are considered like a decision support system to help business user to rewrite user queries. Following the input SPARQL query of the motivating example including a CPV code, a NUTS region (only coordinates) and some numeric values for total value and duration is presented, see Fig. 1. After the process of query expansion a new SPARQL query¹² is built, see Fig. 2. The process of expansion selects new CPV codes (45221100-“Construction work for bridges”, 45221110-“Bridge construction work”, 45221111-“Road bridge construction work”, 45221113-“Footbridge construction work”), new NUTS codes

¹² The URI prefixes of this example come from the “Prefix.cc” service.

```

SELECT * WHERE{
  ?notice rdf:type ppn-def:PublicProcurementNotice .
  ?notice dct:identifier ?id .
  ?notice dct:date ?date .
  ?notice dct:description ?description .
  ?notice ppn-def:hasStatus ppn-def:Active .
  ?notice org:classification <http://purl.org/organizations#SME> .
  ?notice wgs84_pos:lat ?lat .
  ?notice wgs84_pos:lon ?long .
  ?notice ppn-def:totalValue ?totalValue .
  ?amount muo:measuredIn <http://purl.org/weso/units/euro> .
  ?notice ppn-def:duration ?duration .
  ?notice ppn-def:nutsCode ?nutsCode .
  ?duration muo:measuredIn <http://purl.org/weso/units/year> .
  ?notice cpv-def:codeIn2008 ?cpvCode .
FILTER (
  ((?nutsCode = nuts:BE)) and
  ((?cpvCode = cpv:45221000))
  and (?lat == "50.85") and (?long == "43.49")
  and (?totalValue <= 170,000^xsd:double) and (?duration <= 2) )}

```

Fig. 1. Simple SPARQL query

(spreading the geographical scope) and establish a range for the numeric variables according to the historical information available at the database.

Currently we are finishing the process of publishing the PSCs and the information extracted from public procurement notices as linked data. Moreover an information retrieval system¹³ (implemented using Java technologies) is available to test the process of expansion. On the other hand the result set is sorted according to a rank function. This point is ongoing research due to the fact that a lot of OWA operators⁸ and Entity Ranking Functions¹² are available.

4 Performance Evaluation

The evaluation of the system can be carried out from two different points of view: 1) With regards to the validation of the goodness and the improvement of the proposed system we have identified, apart from selecting a service to be tested, three main variables: a) the amount of information used; b) the number of tests (execution of prepared user queries) that should be carried out to assess a correct precision and recall of the proposed retrieval system and c) the best combination of expansion methods. From the first variable point of view 1M public procurement notices (provided by Gateway SCS-Euroalert.net¹⁴) and over 320K organizations¹⁵ are available. On the second one, we have not decided yet

¹³ MOLDEAS-<http://moldeas-web.appspot.com>

¹⁴ <http://euroalert.net/>

¹⁵ <ftp://ftp.ted.europa.eu/META-XML/>

```

SELECT * WHERE{
  ...
  ?notice nuts:containedBy ?place .
FILTER ( ( (?cpvCode = cpv:45221000) or
            (?cpvCode = cpv:45221110) or
            (?cpvCode = cpv:45221111)... )
          ( (?place nuts:containedBy nuts:NL326 ) or
            (?place nuts:containedBy nuts:B3) or
            (?place nuts:containedBy nuts:BE2) or ... )
          and (?duration >= 2 and ?duration <= 4)
          and (?date >= 2008 and ?date <= 2011)
          and (?totalValue > 130,000xsd:double
              and ?totalValue <= 200,000xsd:double))}

```

Fig. 2. Expanded SPARQL query

how many tests would be appropriate to provide a correct evaluation but the information about how many queries are requested per day in the existing public systems can be a right trail. The expected result of this evaluation supposes the first step to validate our approach and select the best combination of expansion methods to improve the access and retrieval of the information about public procurement notices using the LOD approach.

2) In the case of performance evaluation the first tests showed us that the execution of expanded queries involved an extra-time to execute them via SPARQL. Checking existing works in SPARQL optimizations [14][16] and efficient querying of triple stores [15] led us to re-think the process of building expanded user queries trying to improve the execution times. In next section the design of the experiment, the steps to accomplish an improvement in the execution of the SPARQL queries and the results of the tests are presented.

4.1 Design of the Experiment

In Sect. 3 the methods to expand an user query were presented to show how the systems works to generate enhanced queries using the information variables of the public procurement notices. In this experiment [16] we will focus on the next variables: CPV and NUTS codes and the publishing year. The CPV is a taxonomy in which concepts are grouped by a category and identified by a code that indicates their category: “Division” e.g. 01000000 , “Group” e.g. 01100000, “Class” e.g. 01110000, “Category” e.g. 01112000, 01112200, 1112210 or 01112211. On the other hand NUTS is the “Nomenclature of Territorial Units for Statistics” established by Eurostat in order to provide a single uniform breakdown of territorial units. Each code begins with a two-letter code referencing the country,

¹⁶ The complete description of the experiment including all tables of selected queries and execution times is available at:

<http://purl.org/weso/moldeas/papers/caepia2011.pdf>.

which is identical to the ISO 3166-1 alpha-2 and for each EU member country three levels of NUTS codes are established. Finally, the publishing year is just a number indicating when the notice was published. Taking into account the description of these variables the proposed methodology is the next one:

Table 2. Description of the tests and optimization features

Test/ Feature	F_1	F_2	F_3	F_4	F_5	F_6	F_7
T_1	*						
T_2	*		*				
T_3		*					
T_4		*	*				
T_5		*	*	*			
T_6^1 (n CPV codes and m NUTS codes)		*	*	*	*	*	
T_6^2 (\equiv)		*	*	*	*	*	*
T_7^1 (1 CPV code and m NUTS codes)		*	*	*		*	
T_7^2 (\equiv)		*	*	*		*	*
T_8^1 (\equiv)		*	*	*	*	*	
T_8^2 (\equiv)		*	*	*	*	*	*
T_9^1 (1 CPV code and 1 NUTS code)		*	*	*		*	
T_9^2 (\equiv)		*	*	*		*	*
T_{10}^1 (\equiv)		*	*	*	*	*	
T_{10}^2 (\equiv)		*	*	*	*	*	*

- Select the initial CPV codes (with different categories) to build simple and expanded queries.
- Select the initial NUTS codes.
- Establish the publishing years according to the data in the triple store. Currently, public procurement notices from 2008 to 2011 are stored in the database and grouped by the publishing year using named graphs [\[17\]](#).
- Determine the software and hardware environment.
- Select the datasets stored in the database (e.g. CPV-10K concepts, NUTS-8K codes and Public Procurement Notices-1M of notices altogether about 9 million of triples).
- Build and execute via SPARQL simple and expanded queries with the selected information applying the query expansion methods.
- Combine the different SPARQL and algorithm optimizations, see Table [\[2\]](#).
- Log the execution times and establish the number of replies (e.g. 3) to perform the tests.

¹⁷ E.g: <http://purl.org/weso/ppn/2008>

According to these steps, all queries (9) use a range between 2008 and 2011 for the publishing year. The software environment is comprised of a Virtual Box (version 4.0.6) virtual machine (Linux 2.6.35-22-server #33-Ubuntu 2 SMP x86_64 GNU/Linux Ubuntu 10.10, 2GB RAM and 30GB HardDisk) in which a Open Link Virtuoso¹⁸ instance (version 06.01.3127) is installed. The virtual machine is hosted in a DELL PC (same configuration as virtual machine) and a regular internet connection is used to execute the queries.

After that it is necessary to define the possible optimizations (“description”-*ID*) that will configure the features of the tests as Table 2 shows. In this case a distinction between “simple queries”- F_1 . (1 CPV code) and “enhanced queries”- F_2 . (n CPV codes) should be made. Besides there is a list of SPARQL optimizations that can be applied: “LIMIT clause” (value fixed to 10000)- F_3 , “Rewrite SPARQL queries” (following the aforementioned works and making the matching and filtering of the triples from the most specific to general)- F_4 , “Use of named graphs”- F_5 , “Split enhanced queries into simple queries”- F_6 and “Use of an ad-hoc implementation of the Map/Reduce algorithm by Google, with 5 threads to perform the map function and 1 thread to reduce the results of the queries”- F_7 . Taking into account these features the different tests are performed in 3 replies using the arithmetic mean to aggregate the execution times.

4.2 Results and Discussion

During the tests about 5751 SPARQL queries have been performed in order to retrieve the data and the execution times of the queries. These results are processed using “bash” scripts that extract out the statistics and generate a spreadsheet. Regarding the comparison of results, the calculation of the gain ($t_{old}/t_{new} - 1 * 100$) is tackled in two ways depending on the kind of query (simple or enhanced): 1) comparison of test T_1 with T_2 and 2) comparison of test T_3 with tests $T_4...T_{10}^2$.

Results show there is no sensible gain when some optimizations are put in action like F_3 , F_4 and F_5 . In the case of F_3 the use of the “LIMIT clause” fixed to (10000) is not representative due to the results of the triple matching process are previously filtered. “Rewriting queries” F_4 usually involves an improvement in the execution time but maybe the information variables used in these queries does not allow minimize the target dataset while the triple matching process is being ran. Also when “named graphs” F_5 are used the execution time of a single query is obviously lower than one query over all public procurement notices dataset but the number queries to be performed is higher implying a slower execution time. On the other hand, the optimizations F_6 and F_7 bring strong improvements in the execution times of tests T_6^2 , T_7^1 , T_7^2 and T_9^2 . Nevertheless tests T_{10}^1 and T_{10}^2 do not get an improvement in execution time due to the fact that the addition of some features does not guarantee a real gain. One of the highlighted outcomes of this study lies on the detection that the number of CPV codes¹⁹ in a query is related to the execution time (it is about 3 sec. per code)

¹⁸ <http://virtuoso.openlinksw.com>

¹⁹ Adding or removing NUTS codes does not almost change the execution time.

thus the use of only one CPV code in a query improves the process of retrieving public procurement notices. Finally, the use of distributed algorithms is widely accepted and proven when scalability problems appear. In conclusion, taking into account these results the best configuration to improve the execution time of expanded queries lies on splitting them into simple queries (only one CPV code) and use distributed algorithms like Map/Reduce. Nevertheless, other actions in the scope of hardware configuration, caching results [5], use of information variables in the queries with more entropy, etc. could improve the behavior of the system.

5 Conclusions and Future Work

The implementation of these expansion methods is supposed to afford a new way to exploit the information published inside public procurement notices applying advanced algorithms on LOD. Following we highlight the advantages of this approach: 1) decreasing of the information's dispersion; 2) unification of the data models and formats; 3) implicit support to multilingual and multicultural issues; 4) enrichment of the public procurement notices; 5) alignment with the Digital Agenda for Europe; 6) raise awareness on public procurement opportunities among SMEs and 7) deployment of enhanced services on public procurement notices. Regarding the future work, the results of this study are intended to be exploited by a commercial service like Euroalert.net [11] and we are also interested in reporting the results to *The Internal Market and Services Directorate General (DG MARKT) of the European Commission*, *The Information Society and Media Directorate General (DG INFSO) of the European Commission*, the LOD and OGD initiatives among others. On the other hand, the performance evaluation allows us to identify bottlenecks and test the current system at different levels. Now we have used the execution time as target variable to be improved. Nevertheless if the retrieval system is supposed to work off-line (like an alert system of public procurement notices) the execution time should not be a key-factor to deploy a semantic-based platform for e-Procurement in a production environment that takes advantage of the semantic web technologies and the LOD approach. Finally, we are willingness to check the possibility of using other triple-stores and to perform more load, stress, endurance or usability tests following the evaluation points of view presented in Sect. 4.

References

1. Bernstein, A., Kiefer, C., Stocker, M., O.: A sparql optimization approach based on triple pattern selectivity estimation. Technical report, University of Zurich, Department of (2007)
2. Alor-Hernández, G., Gómez Berbís, J.M., Rodríguez González, A., et al.: HYDRA: A Middleware-Oriented Integrated Architecture for e-Procurement in Supply Chains. *T. Computational Collective Intelligence* 1, 1–20 (2010)

3. Bellaachia, A., Amor-Tijani, G.: Enhanced query expansion in english-arabic clir. In: Proceedings of the 2008 19th International Conference on Database and Expert Systems Application, pp. 61–66. IEEE Computer Society, Washington, DC, USA (2008)
4. Berrueta, D., Labra, J.E., Polo, L.: Searching over Public Administration Legal Documents Using Ontologies. In: JCKBSE, pp. 167–175 (2006)
5. Blanco, R., Bortnikov, E., Junqueira, F., Lempel, R., Telloli, L., Zaragoza, H.: Caching search engine results over incremental indices. In: Proceeding of the 33rd ACM SIGIR 2010 Conference, SIGIR 2010, pp. 82–89. ACM, New York (2010)
6. Arenas, M., Buil-Aranda, C., Corcho, O.: Semantics and optimization of the SPARQL 1.1 federation extension. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 201. LNCS, vol. 6644, pp. 1–15. Springer, Heidelberg (2011)
7. Cohen, P.R., Kjeldsen, R.: Information Retrieval by Constrained Spreading Activation in Semantic Networks. *Inf. Process. Manage.* 23(4), 255–268 (1987)
8. Emrouznejad, A., Amin, G.R.: Document similarity: a new measure using owa. In: Proc. of the 6th FSKD 2009, Piscataway, NJ, USA, pp. 186–190 (2009)
9. Hepp, M.: Possible Ontologies. *IEEE-Internet Computing* 1, 90–96 (2007)
10. Leukel, J., Schmitz, V., et al.: Exchange Of Catalog Dat. In: B2B Relationships - Analysis And Improvement
11. Marín, J., Labra, J.: Doing Business by selling free services. In: Ordóñez, P., et al. (eds.) Web 2.0: The Business Model, part 6, pp. 89–102. Springer, Heidelberg (2009)
12. Pound, J., Mika, P., Zaragoza, H.: Ad-hoc object retrieval in the web of data. In: Proceedings of the 19th WWW 2010, pp. 771–780. ACM, New York (2010)
13. Rocha, C., al, D.S.e.: A Hybrid Approach for Searching in the Semantic Web. In: WWW, pp. 374–383 (2004)
14. Schmidt, M., Meier, M., Lausen, G.: Foundations of sparql query optimization. In: Proceedings of the 13th International Conference on Database Theory, ICDT 2010, pp. 4–33. ACM, New York (2010)
15. Yan, Y., Wang, C., Zhou, A.: Efficiently querying rdf data in triple stores. In: Proceeding of the 17th WWW 2008, pp. 1053–1054. ACM, New York (2008)
16. Álvarez, J., Rubiera, E., Polo, L.: Promoting Government Controlled Vocabularies for the Semantic Web: the EUROVOC Thesaurus and the CPV Product Classification System (2008)

Author Index

- Aguilar-Ruiz, Jesús S. 164
Alaíz, Carlos M. 124
Albisua, Iñaki 74
Alonso-Betanzos, Amparo 84
Alonso-Gonzalez, Carlos J. 223
Álvarez, Jose María 494
Álvarez-García, Sandra 483
Arbelaitz, Olatz 74, 413
Arias, Mario 483
Aznar, Fidel 303
- Barranco, Carlos D. 164
Barreiro, Álvaro 403
Barrenechea, Eurne 283
Bermejo, Pablo 54
Bhattacharya, Joydeep 253
Bielza, Concha 145
Biswas, Gautam 223
Bobadilla, Jesús 433
Bolón-Canedo, Verónica 84
Borrajo, Daniel 183
Brisaboa, Nieves R. 483
Bustince, Humberto 283, 373
- Calmeau, Ramón 494
Castrillón-Santana, Modesto 313
Cerquides, Jesus 42
Cleger-Tamayo, Sergio 423
Conradie, Willem 173
Corchuelo, Rafael 443
Cordón, Oscar 293
Corrales, Víctor 393
Cózar, Javier 353
- Damas, Sergio 293
De Baets, Bernard 283
de Haro-García, Aida 64, 104
de la Ossa, Luis 54, 353
de la Rosa, Tomás 183
del Águila, Isabel M. 213
del Castillo, María Dolores 273
del Jesus, María José 263
del Sagrado, José 213
Díaz-Díaz, Norberto 164
Díez-Pastor, José F. 94
- Dorronsoró, José R. 124
Dowe, David L. 1
- Esteva, marc 12
- Fernandez, Javier 373
Fernández, Javier D. 483
Fernández-Breis, Jesualdo Tomás 473
Fernández-Luna, Juan M. 423
Ferrara, Felice 463
Frías, María Pilar 263
- Galar, Mikel 283
García-Olaya, Angel 183
García-Osorio, César 94
García-Pedrajas, Nicolás 64, 104
García-Torres, Jose M. 293
Garrido, Antonio 233
Gómez-Vela, Francisco 164
Gonçalves, Teresa 453
González, Miguel A. 363
González-Rodríguez, Inés 343
Gorretta, Nathalie 333
Griol, David 393
Guijarro, María 323
Guijarro-Berdiñas, Bertha 84, 114
Gurrutxaga, Ibai 74, 413
Gutierrez, Claudio 483
Gutiérrez-Avilés, David 155
- Hernández, Inma 443
Hernández, Jerónimo 134
Hernández-Orallo, José 1
Hernández-Sosa, Daniel 313
Hernando, Antonio 433
Herrera, P. Javier 323
Huete, Juan F. 423
- Iglesias, Ángel 273
Insa-Cabrera, Javier 1
Inza, Iñaki 134
Iovanella, Antonio 243
- Jimenez, Fernando 383
Juarez, José M. 383
Jurio, Aranzazu 373

- Labbé, Sylvain 333
 Labra, José Emilio 494
 Larrañaga, Pedro 145
 Lojo, Aizea 413
 López-Cruz, Pedro L. 145
 Lopez-Molina, Carlos 283
 López-Sánchez, Maite 12
 Lorenzo-Navarro, Javier 313
 Lozano-Tello, Adolfo 203
 Lupiani, Eduardo 383
- Machuca, Enrique 243
 Mandow, Lawrence 243
 Marín, Ángel 494
 Marín, José Luis 494
 Marreo-Estévez, Francisco 22
 Martínez-Álvarez, Francisco 164
 Martínez-Prieto, Miguel A. 483
 Mencía, Carlos 193
 Mikhaylov, Boris 42
 Miñarro-Giménez, José Antonio 473
 Miranda, Nuno 453
 Molina, José Manuel 393
 Morales, Javier 12
 Morales, Lluvia 233
 Moya, Noemi 223
 Muguerza, Javier 74, 413
- Navarro, Fernando 293
- Oliva, Jesús 273
 Orellana, Francisco J. 213
 Ortega, Fernando 433
- Padrón-Ferrer, Antonio 22
 Pajares, Gonzalo 323
 Palma, José 383
 Parapar, Javier 403
 Parras, Manuel 263
 Paternain, Daniel 373
 Pereda, Ernesto 253
 Pérez, Jesús M. 74
 Pérez de la Cruz, José Luis 243
 Pérez-Godoy, María Dolores 263
- Pérez-Recuerda, Pedro 263
 Pérez-Rodríguez, Javier 64, 104
 Pérez-Sánchez, Beatriz 114
 Perona, Iñigo 413
 Peteiro-Barral, Diego 84, 114
 Prieto, Alvaro E. 203
 Puente, Jorge 343
 Puerta, Jose M. 54, 353
 Pujol, Mar 303
- Quaresma, Paulo 453
- Rabatel, Gilles 333
 Raminhos, Ricardo 453
 Redondo-García, José Luis 203
 Reich, Gregor 32
 Reiterer, Susanne 253
 Riquelme, José C. 155
 Rivera, Antonio Jesús 263
 Rivero, Carlos R. 443
 Rizo, Ramón 303
 Rodríguez, Juan J. 94
 Rodríguez-Aguilar, Juan A. 42
 Rodríguez-Baena, Domingo Savio 164
 Rubio-Escudero, Cristina 155
 Ruiz, David 443
- Saias, José 453
 Sánchez-Marño, Noelia 84
 Sánchez-Nielsen, Elena 22
 Santamaría, José 293
 Santos, Pedro 94
 Sanz, José Antonio 373
 Sciavicco, Guido 173
 Seabra, Pedro 453
 Serina, Ivan 233
 Serrano, José Ignacio 273
 Sierra, María R. 193
- Tasso, Carlo 463
- Varela, Ramiro 193, 363
 Vela, Camino R. 343, 363